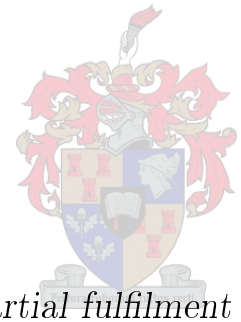
Improving Hyperplane Based Density Clustering Solutions With Applications in Image Processing

by

Jacob Bradley Kenyon



Thesis presented in partial fulfilment of the requirements for the degree of Master of Commerce (Statistics) in the Department of Statistics and Actuarial Science at Stellenbosch University

Supervisor: Dr. David Hofmeyr

April 2019

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: April 2019

Abstract

Improving Hyperplane Based Density Clustering Solutions With Applications in Image Processing

J. B. Kenyon

Department of Statistics and Actuarial Science,
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.

Thesis: MCom (Statistics)

April 2019

Minimum Density Hyperplane (MDH) clustering is a recently proposed method that seeks the location of an optimal low-density separator by directly minimising the integral of the empirical density function on the separating surface. This approach learns underlying clusters within the data in an efficient and scalable way using projection pursuit. The main limitation of MDH is that it defines clusters using a linear hyperplane. In recent research, MDH was applied to data which was non-linearly embedded in a high-dimensional feature space using Kernel Principal Component Analysis. While this method has shown to be an effective approach that extends the linear plane to a non-linear form, it does not scale well. A procedure is needed that can improve the hyperplane solution in an efficient way. We pose a novel approach to improve upon MDH by reassigning observations in a neighbourhood around a hyperplane solution using a gradient ascent procedure, Mean Shift. While Mean Shift is shown to provide promising results, the computation required to reassign objects becomes prohibitive as the size of the dataset increases. To reduce computation, a single step gradient heuristic is proposed whereby observations are reassigned based on the initial gradient evaluated at each point in relation to the hyperplane. This study critically reviews the validity of these approaches through applications with simulated and real-world datasets, with a focus on applications in image segmentation. We show that these approaches have the potential to improve hyperplane solutions.

Opsomming

Verbetering na Minimum Digtheidsbasis-Klustering: 'n Toepassing in Beeldsegmentasie

J. B. Kenyon

Departement Statistiek en Aktuariële Wetenskap,
Universiteit van Stellenbosch,
Privaatsak X1, Matieland 7602, Suid Afrika.

Tesis: MCom (Statistiek)

April 2019

Minimum Digtheid Hipervlak (MDH) tros-vorming is 'n onlangs voorgestelde metode waartydens die optimale ligging van ?n lae digtheids-hipervlak gevind word deur die integraal van die empiriese dightheidsfunksie oor die hipervlak oppervlak te minimimeer. Hierdie benadering maak gebruik van projeksie-najaging om op 'n doeltreffende wyse onderliggende trosse te identifiseer. Die primêre beperking van MDH is dat trosse deur ?n liniêre hipervlak geskei word. In onlangse navorsing is nie-liniêre of kernfunksie gebaseerde hoofkomponent-analise gebruik tydens die toepassing van MDH. Terwyl dit bevind is dat hier-die metode op doeltreffende wyse die liniêre hipervlak uitbrei na 'n nie-liniêre funksie, kan dit nie effektief toegepas word op baie groot datastelle nie. Daar bestaan dus ruimte vir die ontwikkeling van ?n metode om die hipervlakoplos-sing op 'n doeltreende wyse te verbeter. Ons stel derhalwe 'n nuwe benadering voor wat die hertoewysing van datapunte rondom die hipervlak behels, en wat gebruik maak van die ?mean shift gradient ascent? prosedure. Terwyl ons aantoon dat die implementering van die ?mean shift? algoritme belowende resultate lewer, raak die hertoewysing van datapunte te berekenings-intensief namate die grootte van die datastel toeneem. Ten einde die nodige berekeninge te verminder, word ?n meer heuristiese metode voorgestel waarin slegs ?n en-kele stap benodig word. Hiervolgens word waarnemings hertoegewys op grond van die aanvanklike gradiënt van elke punt in verhouding met die hipervlak. In hierdie studie word die geldigheid van bogaande benaderings op datastelle in beeldsegmentering, en op gesimuleerde data, krities beoordeel. Ons toon aan dat die benaderings wel potensiaal het om hipervlak oplossing te verbeter.

Acknowledgements

I would like to express my sincere gratitude to the following people and organisations:

Hannes and Hannelie van Schalkwyk for their invaluable guidance and support.

Stellenbosch University for providing superior facilities and personnel.

Dr. Hofmeyr for his guidance, insight and assistance on the topic of Minimum Density Hyperplanes.

Jürgen Möller for graciously translating the abstract of this thesis into Afrikaans.

University of California, Irvine and University of California, Berkeley for providing data repositories.

Dedications

This thesis is dedicated to a person of great kindness and devotion,
a person who has provided endless support and without
whom none of this would have been possible.
Thank you, Talette Kenyon.

Contents

List of Figures

List of Tables

Chapter 1

Clustering for Image Segmentation

1.1 Introduction

Image segmentation is a field within computer vision that attempts to automatically segment objects in a picture similar to how the human visual system does (Ballard and Brown, 1982). Image segmentation most often forms the initial step for object detection or pattern recognition. An image is represented as a collection of pixels, each containing a measurement of colour or light intensity. There are two common types of images, grayscale and colour. Frequently an image will exhibit a foreground which contains pixels that are more similar to one another than to those contained in the background. Grouping pixels which are similar, provides a logical approach to segmenting an image. Several cluster analysis techniques have been presented as possible solutions for image segmentation.

Cluster analysis is an unsupervised approach that attempts to learn the true underlying class structure within a dataset (Tan *et al.*, 2013). When considering image segmentation, cluster analysis seeks to learn the relational structure of pixel intensities in order to detect patterns within an image. While the human visual system can easily identify objects within an image, the task is notably more difficult for computers. The ability for an application to cluster objects in an automatic way is an important data mining tool which serves as a solution for image segmentation.

There is a plethora of clustering techniques available to analysts. Each technique provides a different approach to grouping and is accompanied with its own set of challenges. The challenges for any given method can be summarised by its constraints, scalability, quality and usability (Tan *et al.*, 2013). The more popular methods embody few constraints, scale well and produce meaningful, useful clusters.

Density-based clustering methods are a popular choice for attempting to solve the task of image segmentation. These techniques are capable of learning complex structures within a dataset. However, most density-based meth-

ods suffer from a lack of scalability. Minimum Density Hyperplane (MDH) clustering is a recent procedure that scales relatively well compared to other density-based approaches. MDH attempts to solve the problem of learning an optimal low-density linear separator via projecting onto a univariate vector. This achieves maximal reduction in dimensionality and results in MDH being more applicable to larger scale problems than other density-based methods. MDH learns an equation which defines the hyperplane solution which allows for deriving solutions from a subset of the data. Other popular density-based methods, such as the well known spatial clustering of applications with noise (Ester *et al.*, 1996, DBSCAN), require a full dataset to obtain a solution and are incapable of sampling to reduce computation. Herein lie some key advantages that MDH has over most other density-based clustering techniques, scalability and its ability to cluster a full dataset using only a subset of the data.

One limiting factor associated with an MDH solution is that the final groupings are determined using a linear hyperplane. Thus, MDH is not able to learn the true underlying class structure of data containing clusters which are not linearly separable. Yates and Pavlidis (2016) presented a method which non-linearly embeds data into a high-dimensional feature space using Kernel Principal Component Analysis (Schölkopf *et al.*, 1998, KPCA). MDH is then applied to the embedded data. The linear separator in the feature space then corresponds to a non-linear surface in the input space. This method has been shown to be effective but can be computationally expensive with a complexity of $\mathcal{O}(n^2)$, where n is the number of observations. This thesis provides a different approach to improving the hyperplane solution, whereby objects in a neighbourhood around the decision boundary are reassigned using gradient ascent. Applying gradient ascent to a region around the low-density separator allows for a final solution which is not constrained by a linear decision boundary and therefore may improve the hyperplane solution.

Mean Shift (Cheng, 1995, MS) is a gradient ascent approach that assigns objects to clusters based on their location within an attraction basin according to an estimated mode of the probability density function underlying the data. Mean Shift has the same time complexity as KPCA, $\mathcal{O}(n^2)$, but since we only consider a subset of the data, the complexity is greatly reduced. Frequently, the gradient ascent trajectory of an observation will not change direction in relation to the hyperplane. We present a heuristic approach to MS based on the initial gradient which further reduces the computation involved with reassigning observations. This procedure calculates the gradient of the probability density function evaluated at a point and reassigns the observation based on the initial gradient in relationship to the hyperplane. If the estimated gradient points towards the hyperplane then it is likely that the gradient ascent trajectory will have converged to a mode opposite the hyperplane and thus signals that the observation should be reassigned to a different cluster.

1.2 Outline

The research undertaken in this thesis is aimed at enhancing the Minimum Density Hyperplane solution by applying a gradient ascent procedure, in the style of MS, to a set of points within a predefined distance to a hyperplane solution. A single step gradient heuristic approach is also evaluated as an attempt to reduce the computational expense involved with MS.

The remainder of this thesis is organised as follows: Chapter 2 outlines cluster analysis, presents a brief survey of clustering techniques, describes a few methods that measure the quality of a clustering solution before concluding with a discussion about various real-world applications. Chapter 3 details Minimum Density Hyperplane clustering and introduces a novel constraint to the location of a hyperplane solution. Chapter 4 describes two methods to improve the hyperplane solution, namely Mean Shift and its heuristic counterpart. These enhancements are evaluated across several benchmark datasets. Chapter 5 illustrates MDH as an image segmentation tool and discusses a pre-processing method that disperses pixels, which can assist with locating an optimal low-density separator. Then an image segmentation comparative study is undertaken using a variety of images, comparing the performance between K-means, Maximum Margin Clustering, MDH and its enhancements. Chapter 6 concludes the thesis with a discussion on the scope and limitations of the proposed enhancements and proposes future research regarding enhancing hyperplane solutions.

Chapter 2

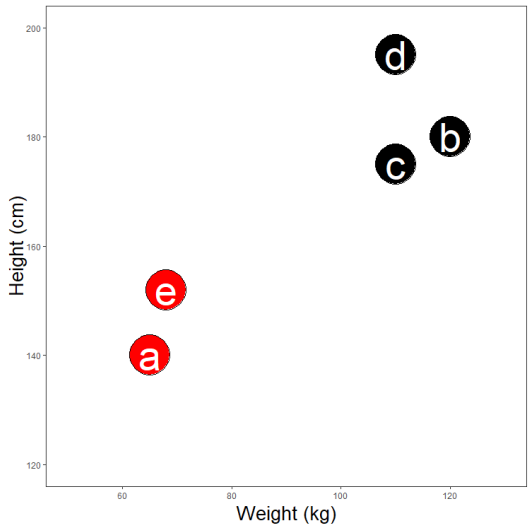
Cluster Analysis

2.1 Introduction

Cluster analysis is an important data mining tool for exploring and understanding information contained within a dataset (Kassambara, 2017). The term cluster analysis (or clustering, data segmentation) refers to a broad set of statistical methods which partition objects into groups which have similar characteristics (James *et al.*, 2013). Clustering is known as an unsupervised machine learning method, since it groups observations within a dataset by learning the composition of clusters without any prior knowledge. Each object can be defined by either a set of measurements or by its relationship to other objects. One common objective of clustering is to divide observations into homogeneous and distinct groups, such that objects within a cluster are more similar to one another than those assigned to other clusters (Hartigan, 1975). There are two fundamental concepts that define the goals of cluster analysis; the notion of *similar* and the meaning of *distinct groups*.

The notion of dissimilarity/similarity is central to determining how observations are grouped within cluster analysis. Similarity is defined as how similar one element is to another and conversely, dissimilarity is defined as how different an element is to another. Often, clustering is based on pairwise dissimilarity measures between objects. One common metric used to define dissimilarity is Euclidean distance (special case of Minkowski distance, L_2norm). Consider a simple example of persons' height and weight (Figure 2.1). As with most 2-dimensional graphical representations, points are represented as a pair of numerical coordinates (x, y), in accordance with the Cartesian coordinate system. Euclidean distances are measurements between points along the Cartesian plane (Equation A.1.1). Since Euclidean distances are defined along the Cartesian plane, it is possible to infer similarities and define clusters by viewing a scatter plot of the data, given the data consists of three or fewer dimensions and the axis measurements are of equal ratio. The ratio of the axis is important since cluster analysis is not scale invariant with respect to

changes of units in a single axis. One should bear this in mind before applying a method that utilises Euclidean distances and consider whether measurements require scaling (Edwards and Cavalli-Sforza, 1965). From the simple example of persons' height and weight, two distinct clusters are evident, one cluster contains subjects a and e while the other group consist of b, c and d.



Subject	Weight	Height
a	65	140
b	120	180
c	110	175
d	110	195
e	68	152

Figure 2.1: Simple example of persons' height and weight.

The choice of distance measure is important since clusters formed by one dissimilarity measurement can be very different than those derived from others. When possible, the choice of dissimilarity metric should be based on pre-existing knowledge of the data (Friedman *et al.*, 2001). As an alternative to grouping observations based on distance metrics, one could define clusters using densities. Density-based techniques define clusters as a group of observations sharing a common estimated probability density mode. Density-based clustering is central to this text and is more aptly detailed in Section 2.3.4. The various dissimilarity/similarity metrics will not be covered in this text and interested readers are referred to Cox and Cox (2000). Attention now turns to the meaning of *distinct groups*.

Carmichael and Julius (1968) defined distinct groups as contiguous, densely populated areas within a dataset which are separated by contiguous relatively empty regions. Figure 2.2 illustrates some examples of distinct groups (a-c) and one which is considered indistinct (d), according to the definition by Carmichael and Julius (1968). For the remainder of this thesis, these distinct data structures are referred to as; *Type A* for linearly separable (Figure 2.2(a)), *Type B* for non-linearly separable (Figure 2.2(b)), *Type C* for dynamically separable (Figure 2.2(c)) and *Type D* refers to overlapping classes (Figure 2.2(d)). While many algorithms seek to divide the data into distinct groups, there are

techniques which allow for overlapping cluster solutions, such as fuzzy clustering. For the purpose of this text, these techniques will not be covered and interested readers are directed to Evers *et al.* (1999) for further details on fuzzy-based clustering methods.

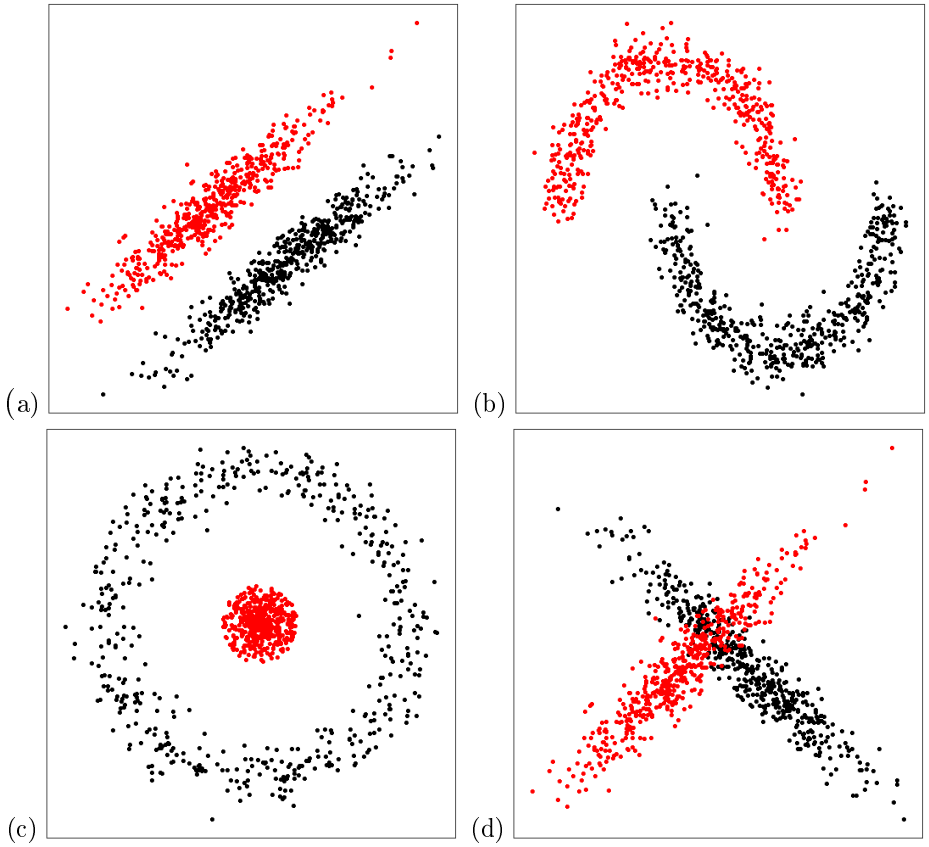


Figure 2.2: Linearly separable (a), non-linearly separable (b), dynamically separable (c) and overlapping group (d) dataset structures.

Cluster analysis is utilised extensively within data mining. It serves two main utilities: it can be used as a pre-processing utility for application in other algorithms and/or as a stand-alone tool to derive insight into the distribution of a dataset. Cluster analysis has rich applications across multiple research disciplines. The goal of this chapter will be to elaborate on these statements while providing general insight into various clustering techniques. The remainder of this chapter is organised as follows: We begin by reviewing the two main purposes of clustering. This is followed by a brief survey of clustering techniques. Afterwards the topic of cluster validation and assessment is discussed. Lastly, a few practical applications in key research disciplines are discussed before concluding the chapter with a summary.

2.2 Main Purposes

Cluster analysis can be summarised as serving one of two typical purposes; stand-alone or an intermediate tool. As an intermediate tool, clustering can be used as a pre-processing step for other algorithms. As a stand-alone tool, cluster analysis is utilised to gain insight into the underlying structure present within data. We begin with a discussion on how cluster analysis can be utilised as an intermediate tool.

2.2.1 Intermediate tool

Clustering is useful as an intermediate step for other data mining tasks such as generating a compact summary of data for classification, hypothesis testing and outlier detection (Tan *et al.*, 2013). Cluster analysis can be utilised as a preprocessing step for other algorithms to reduce computational expense. This section focuses on ways in which cluster analysis can be used as a data reduction technique.

In cluster analysis a group can be characterised according to a cluster prototype or prototypes. A prototype is an object or position within a cluster that is an ideal representation of all other observations within the cluster (Tan *et al.*, 2013). These prototypes are often represented by the mean or medoid of all points within a cluster. The mean is typically used when observations are continuous while the mediod is ideal for discrete categorical data or when the mean cannot be defined. A medoid prototype is an element within a cluster which exhibits the lowest average distance to all other objects within its group (Struyf *et al.*, 1997). Defining the most representative cluster prototypes is useful for multidimensional scaling and as a method to efficiently find nearest neighbours (Friedman *et al.*, 2001).

If a given number of cluster prototypes represent the overall data structure well, then these prototypes can be used as inputs for data modelling. Well positioned prototypes can produce similar results to what the full dataset would have produced (Tan *et al.*, 2013). Utilising a set of cluster prototypes, of size smaller than n, reduces the space and time complexity required by a statistical procedure. For instance, in nearest neighbour applications the pairwise distance between all points is required. Using well positioned cluster prototypes in place of points, reduces the number of distance calculations required. Thus, locating the nearest neighbour prototype for any given object only requires computation of the distance from said object to all prototypes (Friedman *et al.*, 2001).

Clustering provides a meaningful method for dimension reduction when data contain a large number of covariates (d). Gene expression data often contain more covariates than observations, i.e. $d > n$ (Eisen *et al.*, 1998). In this case, multiple linear regression using least squares is not possible since

the number of coefficient estimates exceeds the number of observations (James *et al.*, 2013). Utilising clustering as a dimension reduction technique to reduce the number of covariates by representing them as cluster prototypes (d^*, with $d^* < n$) allows for the application of statistical methods which would otherwise not be possible.

2.2.2 Stand-alone tool

In today's *big data* world, clustering is a valuable stand-alone tool. Clustering plays an important role in online recommendation systems for Amazon, Netflix, and YouTube (Linden *et al.*, 2003). Essentially, cluster analysis seeks like-minded users in order to provide services which will most likely cater to their desires.

Clustering can also provide a framework for a new classification structure. In order to understand a new object or phenomenon, researchers explore the dataset defining said object and compare it with closely related known objects using cluster analysis (Xu and Wunsch, 2005). The hope is that identifying these clusters will increases the overall knowledge and understanding that people will have of this phenomenon in the future. This automatic learning process plays an important role in several fields of research. These include but are not limited to: biology, medicine, psychiatry, economics and multimedia analysis. The role that clustering serves within said fields is further discussed within Section 2.5.

Clustering is also a common tool for spatial data analysis (Halkidi *et al.*, 2001*a*). Spatial data consist of information that identifies various objects such as oceans, naturally occurring and constructed features, commercial and residential zones and socio-economic indicators to name a few (Bailey and Gatrell, 1995). Given the size of such datasets, it is labour intensive and most often infeasible for analysts to manually examine spatial data in detail. Clustering provides an automatic process for analysing data by identifying and extracting useful patterns and characteristics that may exist (Halkidi *et al.*, 2001*a*).

Clustering as a stand-alone application is also popular within computer vision, utilised as a multimedia processing and query technique. In this regard, clustering can be utilised to identify interesting shapes within images, track objects within videos, compress multimedia files to reduce storage requirements and provide a system for fast retrieval of information contained online (Berkhin, 2006). Image segmentation is the process which partitions an image into different segments which contain similar attributes (viz. pixel intensities). It can be considered a pre-processing step also if one is concerned with object recognition, tracking or image analysis (Kumar, 2017). One can segment a single image and then group a collection of segmented images based on similarities which can then decrease the required time to query such information. These systems are popular for online image queries, such as that

utilised within Google's image search engine (Deselaers *et al.*, 2003). Image segmentation using clustering is the topic of Chapter 5. Therein, image segmentation is discussed and illustrated in further detail with an emphasis on enhancing density-based hyperplane solutions.

2.3 Clustering Methods

There is a plethora of clustering methods available to analysts. Each technique may provide different groupings and is accompanied with its own set of challenges. The choice of a particular algorithm is dependent on: desired output, known ability of the method to learn various cluster structures, and the type and size of the dataset in question (Berkhin, 2006). There are two broad classes of clustering, hierarchical and partitioning. Hierarchical techniques sequentially merge observations into clusters (agglomerative algorithms) or divide a dataset into smaller clusters (divisive algorithms). Partitioning methods segment a dataset of n objects into k mutually exclusive clusters. The key difference between partitioning and hierarchical procedures is that hierarchical methods build clusters iteratively while partitioning techniques learn groupings directly.

Clustering methods can be further categorised into five broad *kingdoms*: partitioning, hierarchical, model-based, density-based and grid-based methods (Tan *et al.*, 2013). Some algorithms contain a mixture of these categories and as such some methods can reside within multiple kingdoms. Nevertheless, this scheme of grouping methods is common and assists with discussing attributes of the various clustering techniques.

The challenges for any given method can be summarised by its constraints, scalability, quality, interpretability/usability. The constraints of a technique refers to user-specifications that are required in order to apply a given clustering method. Scalability indicates the efficiency of a technique to obtain a clustering solution from large datasets. The quality of a method is predicated upon its ability to deal with different data types, discover clusters of complex shape and if it is capable of dealing with outliers or noisy data. Interpretability of a clustering technique is defined as whether the method produces meaningful clusters which describes the data well and can be easily understood and used by many people (Aggarwal and Reddy, 2013). The following subsections briefly discuss the various clustering methods and the challenges accompanying each technique.

2.3.1 Partitioning methods

Partitioning methods segment n objects into k groups which optimise a chosen partitioning criterion. There are methods which exhaustively enumerate all partitions seeking the optimal solution and those which apply a heuristic

approach, such as K-means. A common constraint of partitioning methods is that they require the user to pre-define the number of clusters (Xu and Wunsch, 2005). K-means is one of the most popular partitioning-based clustering techniques.

K-means is a technique intended for data which is quantitative where dissimilarities are defined using Euclidean distance and the objective is to minimise within and maximise between cluster variability. The standard K-means algorithm was first presented by Stuart Lloyd in 1957 while working at Bell Labs, which was later published in 1982 (Lloyd, 1982). James MacQueen *et al.* (1967) was the first to coin the term, "K-means". The main concept is to define k centroids, one for each cluster. The procedure begins by randomly/manually selecting k points within the input space of interest. These points represent the initial cluster centroids or prototypes. Then, each observation is assigned to its nearest prototype. Once all objects are assigned to a cluster, a new prototype is calculated for each group. This process repeats until no observations move from one cluster to another.

K-means is well known since it is relatively straightforward and based on the foundation of analysis of variances. An upside to K-means is that it can be an efficient method. The most popular K-means algorithms require $\mathcal{O}(tkn)$ calculations, where t represents the number of iterations, k is the number of clusters and n represents the sample size. These algorithms scale well since normally the number of clusters and iterations required to obtain a clustering is far fewer than the number of observations (Hartigan and Wong, 1979). However, results from K-means strongly depend on the initial points and each solution is based on local optima which tends to be far from the global one (Berkhin, 2006). Furthermore, in some instances empty clusters may be formed from K-means due to poor initialisation. As with all partitioning methods, it is often not obvious what a reasonable value of k should be. Also, outliers can influence cluster structures when the squared error criterion is used (Tan *et al.*, 2013). K-means cannot discover clusters with non-convex shapes and is applicable only when a dataset is continuous Hartigan (1975).

There are several adaptations of K-means to work around the various shortcomings. K-means++, introduced by Arthur and Vassilvitskii (2007) sought to solve the problem of initialisation points. K-modes reduced the impact of noisy data while having the ability to handle categorical data (Huang, 1997). Kernel K-means are partitioning methods brought forth to handle non-convex cluster structures (Dhillon *et al.*, 2004).

The constraint of selecting the number of clusters is a common deterrent for partitioning methods. As with all clustering techniques, the choice of partitioning method will impact scalability, quality and the interpretability of the final results.

2.3.2 Hierarchical methods

Hierarchical clustering methods derive groups through a nested sequence of dividing or combining observations into clusters based on an optimisation criterion and notion of similarity (density or distance-based). Cluster structures derived from hierarchical methods can be represented by dendrograms, hence the name hierarchical clustering. One strength of hierarchical methods is that they do not require a pre-defined number of clusters. One limitation is that iterative refinement cannot be applied to previously constructed clusters (Wu *et al.*, 2008).

There are two main approaches to hierarchical clustering, top-down (divisive) or bottom-up (agglomerative). Divisive algorithms are considered a global approach whereby the initial cluster contains all observations within the dataset, which is then recursively divided into smaller groups until each object is represented by its own group (singleton). Agglomerative algorithms begin with n singleton clusters and then sequentially join two clusters at a time until all objects are represented by one group (Rousseeuw and Kaufman, 1990).

For divisive methods, the user must pre-define a splitting criterion. A common splitting criterion, when using Euclidean distances with quantitative data, is Ward's criterion since it seeks a split which produces the greatest reduction in the sums of squared error (Ward Jr, 1963). The Gini-index is a common criterion when the data contain categorical variables (Fisher, 1995).

For agglomerative methods, criteria for joining objects into clusters are predefined by the user. The choice of criterion is based on the dissimilarity metric within the algorithm, some common types are: single link (nearest neighbour), complete link (diameter), average link (group average) and centroid link (centroid/mean similarity). Each approach defines similarity between clusters differently. Single linkage defines inter-cluster similarity by the pairwise minimum distances between objects from two different groups. Complete linkage and average linkage are similar to single linkage but instead consider maximum distances and average distances. Centroid linkage uses prototype mean distances between clusters to define inter-cluster similarity and combines two clusters that have the smallest centroid distance (Wilks, 2011).

In addition to choosing a splitting or joining criterion, one should specify the number of desired clusters, post hoc. It is obviously not ideal for a cluster to contain only one observation nor the entire dataset. Thus, any desired number of clusters can be obtained by *trimming* the dendrogram to a meaningful number of groups (Tan *et al.*, 2013). In general, hierarchical methods do not scale well to larger datasets and require at least $\mathcal{O}(n^2)$ computations, where n is the number of observations within a dataset. Since hierarchical methods construct clusters in a nested sequence, previous splits and additions cannot be undone. Additionally, it is not always clear which distance metric to apply nor the optimisation criterion to utilise.

Some common hierarchical clustering algorithms which seek to overcome the various shortcomings are: balanced iterative reducing and clustering using hierarchies (BIRCH, Zhang *et al.* (1996)), clustering using representatives (CURE, Guha *et al.* (1998)) and hierarchical clustering using dynamic modeling (CHAMELEON, Karypis *et al.* (1999)). Each of these methods attempt to solve the various challenges posed by hierarchical clustering. CURE is robust to outliers, BIRCH scales linearly which reduces complexity and CHAMELEON can learn complex cluster structures within datasets.

2.3.3 Model-based methods

Model-based methods are a more formal, parametric approach to solving the clustering problem. These methods assume that each cluster is represented by a density which is assumed to be a member of some parametric family, such as Gaussian or Poisson distributions. Consider that $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ defines a dataset consisting of n i.i.d. observations in d-dimensions with joint probability density $p(\mathbf{x})$. The goal is to identify the underlying probabilistic properties of this joint density so as to infer relational structures within the data (Friedman *et al.*, 2001). Model-based methods use a pre-specified mixture of densities, based on what is believed to represent the cluster structure within a dataset. These densities combine to represent the overall underlying probability density function. Then each component's associated parameters are estimated and used to cluster observations based on a pre-defined criterion, such as Bayes' rule (Fraley and Raftery, 1998).

Model-based approaches require k parametric distribution assumptions. The choice of an optimal number of clusters is attenuated by applying parametric tests such as the Bayesian or Akaike information criterion across multiple choices of k (Fraley and Raftery, 1998). Model-based approaches are flexible in the sense that one can choose the distributional composition. These methods are easily understood since they are based upon statistical theory. Model-based approaches can handle missing data and directly model the density of each cluster (Kumar, 2017).

2.3.4 Density-based methods

Density-based methods are the non-parametric alternative to model-based techniques. Clusters are defined as areas of high density separated by regions of low densities. Densities are estimated from a dataset $\mathbf{X} \subset \mathbb{R}^d$ consisting of the realisations of a random variable with an underlying unknown probability density function $p(\mathbf{x})$. The subsequent clusters can be defined in one of two ways; as a set of maximally connected components of level sets $\{\mathbf{x} \in \mathbb{R}^d | p(\mathbf{x}) > \lambda\}$ based on a sensible choice for the level parameter λ (Hartigan, 1975; Rinaldo and Wasserman, 2010) or as a group of observations within the same attraction basin dictated by a probability density function (Azzalini and Menardi, 2014).

An attraction basin is defined as the region for which all observations' gradient ascent trajectories lead to the same estimated mode. Modal regions are defined herein as the area consisting of all observations within an attraction basin sharing a common mode. The benefit of most density-based methods is that a user does not need to pre-specify the number of clusters, such as the case with DBSCAN and Mean Shift clustering (Ester *et al.*, 1996; Cheng, 1995).

DBSCAN is a well known method proposed by Ester *et al.* (1996). A benefit from DBSCAN is that it can identify complex cluster structures within noisy datasets. This technique is robust to outliers as it incorporates a notion of noise, whereby removing points which are considered outliers. Users do not need to specify the number of clusters but must define the minimum amount of observations required for each group, and an ϵ-neighbourhood value. The ϵ-neighbourhood value dictates the window, bandwidth or distance radius around a point to consider when clustering (Ester *et al.*, 1996). One shortcoming of DSCAN is that it does not scale well, with a time complexity of $\mathcal{O}(n^2)$. However, this can be reduced to $\mathcal{O}(nlog(n))$ by using efficient structures found in lower dimensional spaces of the data (Hinneburg *et al.*, 1998).

Mean Shift (Cheng, 1995, MS) is a mode-seeking procedure that utilises gradient ascent based on Gaussian kernels to assign clusters within a dataset. MS is commonly utilised as a stand-alone method for image segmentation, visual tracking, space analysis and mode-seeking (Fukunaga and Hostetler, 1975; Comaniciu and Meer, 1999; Comaniciu *et al.*, 2000). Mean Shift does not assume any prior cluster structure. It can learn non-convex clusters and is robust to outliers. The only requirement is that a user must specify the window size. The choice of window size (Parzen window or kernel bandwidth) represents a trade-off between generality and accuracy. Selecting a window size is not trival and ultimately determines the final output. Larger values increase the overall smoothness of an estimated probability density and can cause modes to merge. Conversely, smaller values decrease smoothness and generate additional *shallow* modes. To mitigate the problem of bandwidth selection, one can use an adaptive window size (Comaniciu *et al.*, 2001). MS has a time complexity of $\mathcal{O}(n^2)$ and does not scale well to larger datasets. However, complexity can be reduced to $\mathcal{O}(n \log n)$ if only neighbouring observations are considered during the computation of MS (Cheng, 1995).

Consider again the example of person's height and weight. Figure 2.3 illustrates how MS clusters the data. The contour lines represent the estimated probability density function. These contours provide a visualisation of the attraction basins. Person d is within the attraction basin dictated by the right-most mode and as such is assigned to the cluster containing persons b, c and d. Each object's gradient ascent trajectory is indicated as green dots. Notice that each object requires a different number of iterations to converge to a mode. Points further away require more iterations than those closer to their respective mode. Data structures with greater variance may require relatively

more iterations than those containing less variability. MS is discussed in more detail in Chapter 4.

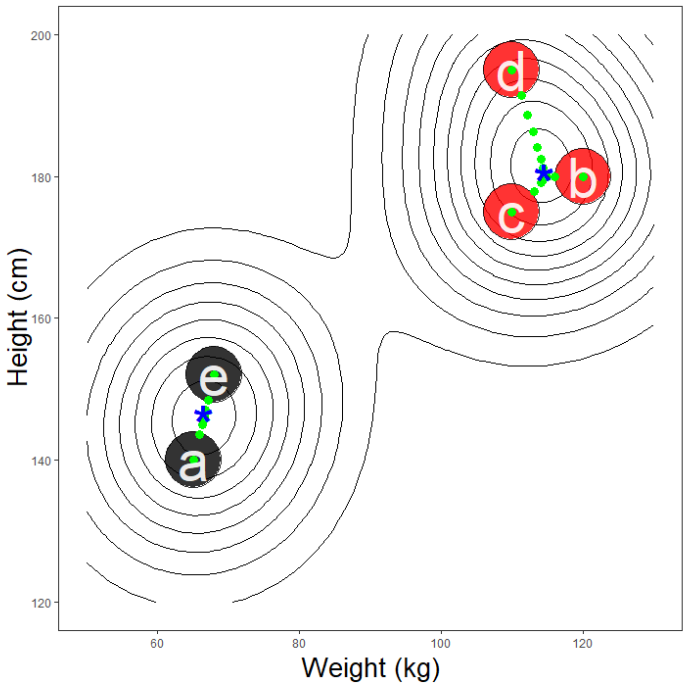


Figure 2.3: Mean Shift solution path from clustering the simple example of persons' height and weight. Green points represent the gradient ascent path of each element towards its associated mode (blue asterisks).

Minimum Density Hyperplane (Pavlidis *et al.*, 2016, MDH) clustering is a recent technique which scales relatively well compared to other density-based methods, such as MS and DBSCAN. MDH clusters observations by directly identifying a low-density separator that partitions at least one dense region from all others, using projection pursuit. The integral of the empirical density function along a hyperplane which is a minimiser represents the low-density separator according to the formulation posed by Ben-David *et al.* (2009). Projection pursuit seeks the linear transformation that results in a highly separable space, whereby the integral of the probability density function along the hyperplane is minimised. Maximal dimension reduction is achieved by projecting onto a vector. This is particularly advantageous for high-dimensional problems. Since MDH yields a binary partition, it cannot learn more than two clusters from a dataset. Minimum Density Divisive Clustering (Hofmeyr and Pavlidis, 2018, MDDC) is an extension of MDH which is capable of clustering more than two groups by creating a hierarchical collection of hyperplanes. Also, since MDH defines clusters using a hyperplane, it is incapable of learning cluster structures which are not linearly separable. In this thesis we formulate

a technique to overcome this limitation by applying a gradient ascent procedure to a collection of observations around the hyperplane solution. MDH is discussed in further detail in Chapter 3 and the proposed enhancement to the hyperplane solution are detailed in Chapter 4.

Figure 2.4 illustrates how MDH assigns the simple example of person's height and weight to clusters, using a univariate estimated density function along the direction orthogonal to the minimum density hyperplane, which is indicated as a green line. Observations that lie below the hyperplane solution are assigned to the red cluster (a and e). Conversely, those above the plane are assigned to the black cluster (b, c, and d).

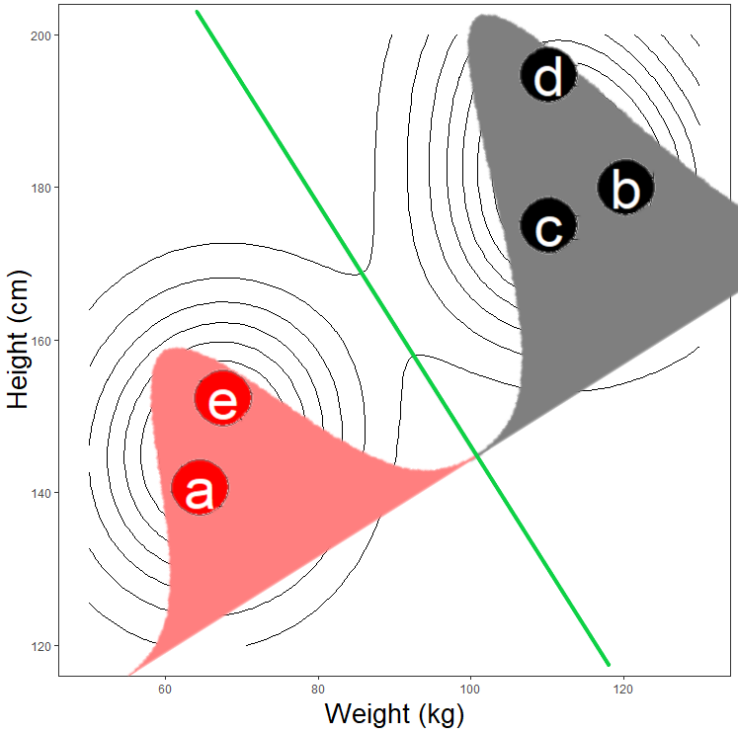


Figure 2.4: MDH solution of the simple example of persons' height and weight. The area under the density is coloured to match each assigned cluster and the low-density separator is indicated as a green line.

2.3.5 Grid-based methods

Grid-based clustering is commonly used for data query operations. These methods divide a dataset into a finite number of cells that form a grid structure. Clustering techniques are then applied to the cells within the grid structure. These methods are scalable since typically the number of cells is far fewer than the number of data points. One setback to using grid structures is the difficulty in capturing irregular cluster structures within the data across cells.

Also, these methods may suffer from local optima due to user-defined cell sizes, borders and density thresholds. Thus, the choice of cell structures crucially impacts final results. Also, grid-based methods generally perform poorly given high dimensional data (Aggarwal and Reddy, 2013).

One popular grid-based method is known as statistical information grid approach, STING (Wang *et al.*, 1997). STING splits the input space into grids in a divisive hierarchical way, where the first layer contains few cells and subsequent layers have an increasing number of cells. Parameters defining observations within each cluster are then stored within each cell and used to answer queries. Parameters at higher levels are easily determined by the information contained within its nested levels. STING is an efficient algorithm with $\mathcal{O}(c)$ complexity since only relevant cells (c) are recursively explored when processing a query (Wang *et al.*, 1997). Another popular grid-based method is known as clustering in quest, CLIQUE. Interested readers are directed to Agrawal *et al.* (1998) for further details.

2.4 Cluster Validation And Assessment

Cluster validation and assessment are as diverse as the topic of clustering. Thus, there exist many forms to assess a clustering method. There are three main tasks which cluster validation seeks to address; tendency, stability and assessment. As an initial step, one should evaluate whether a dataset has any underlying cluster structure and if it is even suitable to apply clustering to the data in the first place. This is known as cluster tendency. Cluster stability evaluates how sensitive a clustering method is to a given set of parameters. Cluster assessment attempts to measure the quality of a clustering solution. Given the multitude of different cluster definitions, there is no common method to assess all clustering solutions. In some instances, cluster validation is subjective to the user, such is often the case when evaluating image segmentation results. Nevertheless, there are three broad categories for measuring the quality of a clustering algorithm; external, internal and relative measures (Zaki *et al.*, 2014). The following section presents the notation used in the remainder of this chapter. This is followed by outlining cluster stability, tendency and a discussion of various cluster validation measurements.

2.4.1 Notation

For clarity moving forward, consider that $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{n}$ defines a dataset in d-dimensions consisting of n observations partitioned into k classes. Furthermore, let $y_i \in \{1, 2, ..., k\}$ signify the true cluster labels (ground truth) for each point. Let the ground truth partitions be defined as $\mathcal{T} = \{T_1, T_2, ..., T_k\}$, where the cluster T_j encapsulates all points with label j, thus $T_j = \{x_i \in \mathbf{X} | y_i = j\}$. Clusters defined by a given method will be denoted, $\mathcal{C} = \{C_1, C_2, ..., C_r\}$ with

$\hat{y}_i \in \{1, 2, ..., r\}$ representing the assigned labels of $\mathbf{x}_i$. When discussing external measures, these metrics rely on an $r \times k$ contingency table ($\mathbf{N}$) formed by comparing a clustering ($\mathcal{C}$) with the ground truth ($\mathcal{T}$) defined as:

$$\mathbf{N}(i, j) = n_{ij} = |C_i \cap T_j|, \tag{2.4.1}$$

where n_{ij} denotes the common number of objects between the cluster C_i and the ground truth T_j. The computational complexity of generating results from this contingency table is $\mathcal{O}(n)$, since evaluating each pair, $\hat{y}_i$ and y_i, from $\mathbf{x}_i \in \mathbf{X}$ corresponds to incrementally adjusting the count n_{ij} (Zaki *et al.*, 2014). Furthermore, take notice that the number of true clusters, k, is distinguished from that which is produced by a clustering algorithm, r.

2.4.2 Cluster tendency and stability

Cluster tendency assesses whether the data has any inherent grouping structure. It is difficult to assess given the various ways in which each clustering method defines a cluster. Nevertheless, a few common assessment techniques are spatial histograms, distance distribution and the Hopkins statistic Zaki *et al.* (2014); Halkidi *et al.* (2001*b*).

Cluster stability relies on the premise that clusters obtained from several bootstrapped datasets should be similar or *stable*. It is typically utilised to determine the optimal number of clusters. The approach begins by taking B bootstrap samples from $\mathbf{X}$ of size n with replacement. For each of B bootstrap samples, apply the clustering method for every r value. Then the distances between all pairs of clustering $C_r(\mathbf{X}_i)$ and $C_r(\mathbf{X}_j)$ are computed to estimate the expected pairwise distance for each value of r. Ultimately, the value of r that exhibits the least amount of variation between the bootstrapped clusters is chosen, as it represents the most stable choice (Zaki *et al.*, 2014).

When considering stability in the context of low-density hyperplanes, there exists a significant relationship. Essentially, a hyperplane solution associated with a lower integrated density is considered more stable than those with relatively higher associated integrated densities (Ben-David and Von Luxburg, 2008). Those hyperplane solutions which are associated with relatively larger integrated densities tend to yield highly variable solutions compared to hyperplane solutions with lower densities.

2.4.3 External measures

External cluster validation compares a clustering solution $\mathcal{C}$ against an *externally* provided set of ground truth labels, $\mathcal{T}$. External measures are supervised validation techniques that require some prior knowledge of the class structure within a dataset. External measures are utilised to gain insight into

how well a given algorithm recovers the known class structure (Kassambara, 2017). Quality clustering methods are considered as those which produce pure, homogeneous clusters which assign observations to their true class, cluster completeness. Additionally, to include a heterogeneous element into a pure cluster should be penalised more than if it was clustered into a *miscellaneous* or less homogeneous category. Also, dividing a relatively smaller sized cluster into further segments is considered more harmful than splitting a cluster containing a relatively greater number of observations (Zaki *et al.*, 2014). Some commonly used external validation metrics include: matching-based, entropy-based, pairwise and correlation measures. A more recent external measure known as Success Ratio was proposed by Pavlidis *et al.* (2016), which captures the binary partition performance of a given clustering method.

2.4.3.1 *Success Ratio*

The Success Ratio is a metric which expresses how well a clustering algorithm groups data into two clusters. While this method is ideal for two-class datasets, it can be extended to data containing more than two classes. In such scenarios, when k is greater than two, labels are aggregated by assigning each element to the group which contains the majority of its members. The Success Ratio indicates how distinct the majority of at least one cluster is from the rest of the data. Success ratio values of zero indicate that an algorithm failed to locate the majority of any cluster from within the data. Larger values signal a better quality of clustering. Values of one indicate that all clusters remain intact after the binary partition. Thus, a good binary partitioning method is one that clusters at least one class to a group that is distinct from all others within the data (Pavlidis *et al.*, 2016).

To calculate the Success Ratio when k is greater than 2, the true class labels are aggregated, denoted $\mathcal{T}_1^*$ and $\mathcal{T}_2^*$. Then, the binary partition error $E(\mathcal{C}_1, \mathcal{C}_2)$ is calculated as defined in Equation 2.4.1. Essentially, the binary partition error represents the number of objects which are not grouped with the cluster containing the majority of its true class members. In contrast, the success of a cluster, $S(\mathcal{C}_1, \mathcal{C}_2)$, captures the extent to which observations are grouped to a cluster containing the majority of their original class members (Equation 2.4.2). Overall, the Success Ratio ($SR(\mathcal{C}_1, \mathcal{C}_2)$) measures the capability of a clustering technique to cluster the data such that it does not separate observations belonging to the same class (Pavlidis *et al.*, 2016). The Success Ratio is defined as:

$$E(\mathcal{C}_1, \mathcal{C}_2) = min\Big\{|\mathcal{C}_1 \cap \mathcal{T}_1^*| + |\mathcal{C}_2 \cap \mathcal{T}_2^*|, |\mathcal{C}_1 \cap \mathcal{T}_2^*| + |\mathcal{C}_2 \cap \mathcal{T}_1^*|\Big\} \tag{2.4.2}$$

$$S(\mathcal{C}_1, \mathcal{C}_2) = min\Big\{max\big\{|\mathcal{C}_1 \cap \mathcal{T}_1^*|, |\mathcal{C}_1^* \cap \mathcal{T}_2^*|\big\}, max\big\{|\mathcal{C}_2 \cap \mathcal{T}_1^*|, |\mathcal{C}_2 \cap \mathcal{T}_2^*|\big\}\Big\} \tag{2.4.3}$$

$$SR(\mathcal{C}_1, \mathcal{C}_2) = \frac{S(\mathcal{C}_1, \mathcal{C}_2)}{S(\mathcal{C}_1, \mathcal{C}_2) + E(\mathcal{C}_1, \mathcal{C}_2)} \tag{2.4.4}$$

2.4.4 Internal measures

Internal measures are unsupervised methods which assess the quality of a clustering method based solely on the data. Internal metrics attempt to capture the extent of how similar objects are within clusters (cluster compactness) and the dissimilarity between clusters (separation). Intra-cluster compactness and inter-cluster separation are obtained via pairwise distance measurements defined by $\delta(\mathbf{x}_i, \mathbf{x}_j)$, which is assumed to be the Euclidean distance between $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}$ (Zaki *et al.*, 2014). When possible, the choice of dissimilarity matrix used to calculate the quality of clustering should coincide with that used to define the clusters. Overall, internal metrics produce a trade-off between maximising intra-cluster compactness and inter-cluster separation (Halkidi *et al.*, 2001*b*; Zaki *et al.*, 2014).

These metrics are motivated by defining a *good* clustering algorithm as one which produces groupings with high intra-class similarity and high inter-class dissimilarity. One commonly used internal cluster validation measure is the silhouette coefficient (Kassambara, 2017).

2.4.4.1 *Silhouette coefficient*

The silhouette coefficient measures how similar objects are within a cluster and the separation between groups. It captures this as a ratio of how close a point is to all other objects within its cluster and how far it is to those points in a neighbouring cluster. The overall silhouette coefficient is the average of each observation's coefficient s_i which is calculated as:

$$s_i = \frac{\mu_{out}^{min}(\mathbf{x}_i) - \mu_{in}(\mathbf{x}_i)}{max\big\{\mu_{out}^{min}(\mathbf{x}_i), \mu_{in}(\mathbf{x}_i)\big\}}, \tag{2.4.5}$$

where $\mu_{in}(\mathbf{x}_i)$ represents the mean distance between $\mathbf{x}_i$ and all other observations within its cluster $\mathcal{C}_{\hat{y}_i}$ and defined as:

$$\mu_{in}(\mathbf{x}_i) = \frac{\Big(\sum_{\mathbf{x}_j \in \mathcal{C}_{\hat{y}_i}, j \neq i} \delta\big(\mathbf{x}_i, \mathbf{x}_j\big)\Big)}{n_{\hat{y}_i} - 1}, \tag{2.4.6}$$

and $\mu_{out}^{min}(\mathbf{x}_i)$ represents the mean distances between $\mathbf{x}_i$ and all other objects within the closest cluster, defined as:

$$\mu_{out}^{min}(x_i) = \min_{j \neq \hat{y}_i} \frac{\sum_{y \in \mathcal{C}_j} \delta\big(\mathbf{x}_i, y\big)}{n_j} \tag{2.4.7}$$

The overall silhouette coefficient is thus defined as:

$$SC = \frac{1}{n}\sum_{i=1}^{n} s_i \tag{2.4.8}$$

Silhouette coefficient (s_i) values range from -1 to 1. Negative values indicate that $\mathbf{x}_i$ is closer to another cluster than to observations within its cluster. Positive values indicate that $\mathbf{x}_i$ is relatively far from objects contained within a neighbouring cluster compared to the distance to other observations within its own cluster. In other words, large negative values indicate that $\mathbf{x}_i$ is possibly assigned to the wrong cluster. Zero indicates that $\mathbf{x}_i$ lies between two clusters and values close to 1 indicate that $\mathbf{x}_i$ is well clustered. The silhouette coefficient average (SC) is interpreted similarly but with regard to all observations, with values close to one indicating a quality clustering solution (Zaki *et al.*, 2014).

2.4.5 Relative measures

Relative measures are utilised to gain insight into the performance a specific clustering method exhibits given various parameter settings (Zaki *et al.*, 2014). For example, comparing solutions from K-means applied to a dataset with k set to 2, 5, or 10. Some common metrics are: silhouette coefficient, gap statistic and Calinski-Harabasz index.

We briefly mentioned that there exists a relationship between a low-density separator and cluster stability. Herein we propose a novel approach, for future research, which attempts to select an optimal number of k for Minimum Density Divisive Clustering (MDDC). In this approach, one could apply MDDC to a dataset using various values of k. Then for each MDDC solution, the average proportion of observations within a neighbourhood of each low-density separator relative to the entire dataset is calculated. To implement this approach one would have to apply a penalty term since the average proportion over all low-density neighbourhoods will most likely exhibit a monotone decreasing characteristic as the value of k increases. The k value associated with the MMDC solution with the lowest penalised average proportion of observations within the hyperplanes' locations is considered stable and is posed as an optimal value for k. Since this thesis focuses on improving a single hyperplane solution, the above proposed method is left for future research.

Technical details regarding relative measures are omitted and readers are referred to Zaki *et al.* (2014) and Halkidi *et al.* (2001*b*) in this regard. The focus now shifts to discussing real-world applications of clustering to elaborate its usefulness and prominence as an analytic tool.

2.5 Broad Applications of Cluster Analysis

Given that cluster analysis is the study of how similar objects are, it provides insight to a bevy of problems encompassing a wide range of domain applications. This section briefly discusses some of the various disciplines that utilise clustering and the impact thereof.

2.5.1 Biology

Early research in biology has provided a well known science of classifying all living things. Taxonomy is the science of classifying living things into a hierarchical structure: kingdom, phylum, class, order, family, genus, and then species (Sokal, 1963). Moving from kingdom to specie, the degree of similarity within a subclass increases. Applying cluster analysis can provide researchers with insight of a possible class to assign an unknown specie to. In more recent genetic research, clustering was utilised to detect gene expressions which exhibit similar functions. It is common practice in genetics to display relationships using dendrograms attached to heat maps. This provides biologists with insight into underlying gene expressions while simultaneously evaluating similarities (Eisen *et al.*, 1998).

2.5.2 Medicine

Medicine is used throughout the world to treat patients with illnesses or specific conditions. Cluster analysis can be used to identify and diagnose patient's conditions. For instance, Ramaswamy *et al.* (2001) clustered tumour gene expressions using an average linkage hierarchical method to assist in the diagnosis of cancer types within patients. Moore *et al.* (2010) utilised additive hierarchical clustering which led to the determination that new classification methods are necessary to diagnose the severity of asthma within patients. Thus, clustering can often be utilised as a tool to design future classification structures. The method of image segmentation is often used within Medicine. Applying cluster analysis to an image can assist with detecting abnormalities within medical scans and aid with diagnosing illnesses.

2.5.3 Psychiatry

Psychiatry is a field of medical research concentrated on the diagnosis, prevention and treatment of mental disorders (Gelder *et al.*, 1989). Clustering has been utilised to discover different types of depression and the causes thereof, usually through learning patterns in longitudinal data. Gould *et al.* (1994) utilised clustering to evaluate the suicidal behaviour in New Zealand. Ellegood *et al.* (2015) performed hierarchical clustering which identified previously unknown connections between neuroanatomical similarities within autistic sub-

jects. Those are just a few examples of the many applications of cluster analysis used to assist in determining different causes and types of mental disorders.

2.5.4 Economics

Applications of cluster analysis can benefit business decisions, especially with regard to marketing strategies. A key objective in marketing research is to identify groupings of similar persons and/or products so as to optimise advertisement placement (Punj and Stewart, 1983). Market segmentation can provide insight into consumer behaviour based on similarities shared within a group and assist companies with product placement decisions. From an economic perspective, cluster analysis has been implemented in an attempt to measure the welfare and quality of life across groups of people based on their location and various demographic information (Hirschberg *et al.*, 1991).

2.5.5 Multimedia

Multimedia data contain images, video, sound or a mixture of each. It is sometimes desired to reduce the storage size of such content. Compression is a utility based on clustering, whereby storage size is reduced by representing observations with their associated prototype value. This is commonly applied to media data where substantial reduction in data size is desired for storage purposes and the loss of some information is deemed acceptable (Tan *et al.*, 2013). One common form of compression is known as vector quantization. Vector quantization creates a table of prototypes where each position within said table is assigned an integer value representing the neighbourhood of a local prototype. Each object is then represented by the index of the prototype representing its cluster (Gersho and Gray, 1991).

While clustering is one solution for multimedia data storage problems, it can also be utilised as a pattern recognition tool. Pattern recognition amounts to identifying shapes within an image. In most cases, the primary objective is to locate objects or detect the edges of an object within an image (Evers *et al.*, 1999). Image segmentation is a form of pattern recognition which seeks to automatically separate objects in a picture similar to how the human visual system does (Parker, 2010). A practical application of image segmentation is explored in detail in Chapter 5.

2.6 Summary

It is clear that cluster analysis is an important statistical tool for applications within data mining. Clustering is a flexible method which can be used as a stand-alone or pre-processing method. However, there is currently no single

approach which can solve all clustering problems, but if one has deep knowledge of the problem task and a broad understanding of the various clustering techniques, a method can be applied which will yield meaningful and useful information.

Measuring the performance of clustering is a difficult task. In some instances, cluster validation is purely subjective. This is often the case with image segmentation. Given a proper measure of clustering quality, cluster validation can provide insight as to how well a method performs compared to other techniques. One can also determine an optimal number of clusters to specify within a clustering procedure using validation metrics.

We have briefly discussed real-world applications of cluster analysis. Not only can cluster analysis provide a structure for future classification rules, it also can assist researchers and doctors with solving critical problems. In conclusion, clustering analysis is an important data mining tool which can provide valuable insight of an underlying structure in data across a wide variety of disciplines.

Chapter 3

Minimum Density Hyperplanes

3.1 Introduction

There is an abundance of density-based clustering methods available to analysts. This chapter focuses on a recently developed technique, known as Minimum Density Hyperplane clustering, created by Pavlidis *et al.* (2016). This approach is based on the problem of learning an optimal low-density linear separator, as proposed by Ben-David *et al.* (2009). This method adopts the density-based clustering definition given by (Hartigan, 1975), whereby a *high-density* cluster is defined as a connected region surrounding a mode of a probability density function. Thus, any points that fall within the same connected high density region are considered to belong to one cluster. A direct consequence of this definition is that clusters are separated by regions of low-density, in accordance with the *low-density separation assumption* (Ben-David *et al.*, 2009).

Minimum Density Hyperplane clustering directly identifies low-density separators by learning the equation of a hyperplane which will partition the modes associated with high-density regions using projection pursuit. Projection pursuit is a class of statistical techniques which seeks an optimal linear transformation, from all possible transformations, that identifies an *interesting* low-dimensional projection of a dataset (Huber, 1985). MDH initialises projection pursuit using a well known method, Principal Component Analysis (PCA). The computation for PCA reduces to the eigen-decomposition problem for a positive semi-definite matrix (*e.g.* covariance matrix) whereby the first principal component is defined along the direction explaining the most variation within the data (Jolliffe, 2011).

The MDH objective seeks an optimal univariate projection upon which a linear decision boundary separates at least one dense region from all others (Hofmeyr and Pavlidis, 2018). Principal components have been used in an attempt to solve this problem (Tasoulis *et al.*, 2010). MDH seeks to improve upon this solution by utilising projection pursuit to further enhance the

separation of at least one dense region from others. A key advantage of using univariate projections is that it renders the density separation problem along the projection almost trivial. Projection pursuit can thus be performed efficiently, making this approach applicable to much larger datasets than is common for most density-based methods (Pavlidis *et al.*, 2016).

The probability density function of a univariate projection within MDH is estimated using a Gaussian kernel density estimator (Pavlidis *et al.*, 2016). A benefit from using Gaussian kernels in the full dimensional space is that it allows one to evaluate the density on a hyperplane exactly using univariate Gaussian kernels. Also, one can reduce time complexity by applying MDH to a subsample of the dataset. The decision rule obtained from the sample of the data can then be applied to the full dataset to obtain a complete clustering solution.

Central to MDH is the choice of a kernel bandwidth to use when estimating the density of the univariate projections. As with most density-based clustering methods, the choice of kernel bandwidth influences the solution and is not trivial. Larger bandwidth values can cause modes to merge, increasing the possibility of not locating an optimal low-density separator. Alternatively, smaller values generate additional *shallow* modes, increasing the possibility of locating a solution plane that does not effectively separate clusters. Whether h is set relatively large or small, the probability density estimate will always exhibit low densities along the boundaries. To mitigate against the possibility of defining a low-density separating hyperplane that is located near the edge of the density function, MDH applies a penalty term during the optimisation process which restricts the hyperplane solution's distance from the mean.

The main limitation of MDH is that it defines a cluster based on a linear hyperplane. While some real-world applications involve datasets which are well separated by linear hyperplanes, there are instances in which clusters cannot be separated linearly (Hofmeyr and Pavlidis, 2018). Yates and Pavlidis (2016) proposed a method to remove the limitation of a linear hyperplane by embedding the data, non-linearly, into a high-dimensional feature space using Kernel Principal Component Analysis. MDH is then applied to the embedded data, where the low-density separator in the feature space corresponds to a non-linear hyperplane in the input space. We propose an alternative approach to remove the limitation imposed by a linear separator. We suggest collecting observations in neighbourhood around a hyperplane solution and then reassigning these observations with a more flexible clustering method. It is thought that by removing the limitations of a linear separator, we can improve hyperplane solutions when the data are non-linearly separable. This is the topic of Chapter 4, wherein two techniques are presented to perform such a task, viz. Mean Shift and a single step gradient heuristic.

When a dataset is thought to consist of more than two clusters, then a single binary partition is not ideal. To overcome this, one can combine several hyperplane solutions in a hierarchical way. This approach is embodied in a

method known as Minimum Density Divisive Clustering (MDDC) (Hofmeyr and Pavlidis, 2018). While the focus of this thesis is on the refinement of a single hyperplane solution, these enhancements can also be applied to each hyperplane during the divisive partitioning of MDDC. Details of MDDC are beyond the scope of this study.

The remainder of this chapter is organized as follows. Formulation and notation are established in Section 2 before applying MDH to the four distinct cluster type datasets (Figure 2.2) in Section 3. The effects of various MDH parameter settings are explored within Section 4 before summarising the chapter in Section 5.

3.2 Formulation

The formulation of MDH (Hofmeyr and Pavlidis, 2018; Pavlidis *et al.*, 2016) sets out to define a hyperplane which bi-partitions a finite dataset, $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^d$, that is assumed to be independent and identically distributed with an unknown probability density function $p(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^+$. A hyperplane is defined as $\mathrm{H}(\boldsymbol{v}, b) := \{\mathbf{x} \in \mathbb{R}^d | \boldsymbol{v}^\top \mathbf{x} = b\}$. The hyperplane $\mathrm{H}(\boldsymbol{v}, b)$ partitions $\mathbf{X}$ into two clusters to which each element is assigned according to the following rule:

$$\mathbf{X} = \mathbf{X}^+_{\boldsymbol{v},b} \cup \mathbf{X}^-_{\boldsymbol{v},b}, \tag{3.2.1}$$

$$\mathbf{X}^-_{\boldsymbol{v},b} := \Big\{\mathbf{x} \in \mathbf{X} \Big| \boldsymbol{v}^\top \mathbf{x} < b\Big\}, \tag{3.2.2}$$

$$\mathbf{X}^+_{\boldsymbol{v},b} := \Big\{\mathbf{x} \in \mathbf{X} \Big| \boldsymbol{v}^\top \mathbf{x} \geq b\Big\}, \tag{3.2.3}$$

where the decision boundary between clusters is defined by the linear equation $\boldsymbol{v}^\top \mathbf{x} = b$. Furthermore, the *projection vector* ($\boldsymbol{v}$) defining the hyperplane, is restricted to have unit norm, where $\boldsymbol{v}^\top \mathbf{X} = \{\boldsymbol{v}^\top \mathbf{x}_i\}_{i=1}^n$ denotes the projection of $\mathbf{X}$ onto $\boldsymbol{v}$.

The *density on the hyperplane* is defined as the integral of $p(\mathbf{x})$:

$$I(\boldsymbol{v}, b) := \int_{H(\boldsymbol{v},b)} p(\mathbf{x}) dx, \tag{3.2.4}$$

where $p(\mathbf{x})$ is approximated by an isotropic Gaussian kernel density estimator:

$$\hat{p}(\mathbf{x}\big|\mathbf{X}, h^2\mathbf{I}) = \frac{1}{n(2\pi h^2)^{\frac{d}{2}}} \sum_{i=1}^n \exp\left\{-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2h^2}\right\}. \tag{3.2.5}$$

One of the benefits of estimating $p(\mathbf{x})$ using isotropic Gaussian kernels is that it allows one to evaluate the density on the hyperplane exactly from one-dimensional projections (Pavlidis *et al.*, 2016). The density on the hyperplane

is evaluated by

$$\hat{I}(\boldsymbol{v}, b|\mathbf{X}, h^2\mathbf{I}) := \int_{H(\boldsymbol{v},b)} \hat{p}(x|\mathbf{X}, h^2\mathbf{I})dx, \tag{3.2.6}$$

$$= \frac{1}{nh\sqrt{2\pi}} \sum_{i=1}^{n} \exp\left\{ -\frac{(b - \boldsymbol{v}^\top \mathbf{x}_i)^2}{2h^2} \right\}, \tag{3.2.7}$$

$$= \hat{p}\Big(b|\{\boldsymbol{v}^\top \mathbf{x}_i\}_{i=1}^n, h^2\Big). \tag{3.2.8}$$

To elaborate, consider that the density on $\mathrm{H}(\boldsymbol{v}, b)$ is estimated using Gaussian kernels with a given bandwidth (h) then the density on a hyperplane is given by $\hat{I}(\boldsymbol{v}, b)$. The density on a hyperplane can be evaluated by projecting $\mathbf{X}$ onto $\boldsymbol{v}$, estimating the kernel density using the same bandwidth (h) and evaluating the hyperplane at b.

It is inevitable that the surface integral, $\hat{I}(\boldsymbol{v}, b)$, of $\hat{p}(\mathbf{x})$ on $\mathrm{H}(\boldsymbol{v}, b)$ will approach zero given a relatively large absolute value for b, resulting in a solution that partitions all but a few observations to one cluster (Pavlidis *et al.*, 2016). That is to say, for any $\boldsymbol{v}$, $\lim_{b\to\pm\infty} \hat{I}(\boldsymbol{v}, b) = 0$. To guard against obtaining a solution hyperplane near the boundaries of the data, a penalty term is introduced to $\hat{I}(\boldsymbol{v}, b)$ which constrains the hyperplane's distance from the mean of the data. Given $\Phi\big(\boldsymbol{v}|\mathbf{X}\big)$ is the *projection index*, the optimisation is defined as:

$$\min_{\boldsymbol{v}} \Phi\big(\boldsymbol{v}|\mathbf{X}\big) = \min_{b\in\mathbb{R}} \Big\{ \phi(\boldsymbol{v}, b|\mathbf{X}) \Big\}, \tag{3.2.9}$$

$$\phi(\boldsymbol{v}, b|\mathbf{X}) = \hat{I}(\boldsymbol{v}, b) + C \max\left\{0, -\alpha\sigma_{\boldsymbol{v}} - b, b - \alpha\sigma_{\boldsymbol{v}}\right\}^{1+\epsilon}, \tag{3.2.10}$$

for any constant value C and $\epsilon \in (0, 1)$, where $\sigma_{\boldsymbol{v}}$ represents the standard deviation of $\boldsymbol{v}^\top\mathbf{X}$ and α is a factor which manipulates the overall size of the feasible region. As a final step, one can further restrict the location of the hyperplane solution, post-optimisation, by requiring it to reside between *prominent* modes. This effectively diminishes the possibility of obtaining a poor hyperplane solution due to a relatively large α value. Note that, since this constraint is done after the optimisation process, the differentiability properties of MDH are maintained. The resulting point on $\boldsymbol{v}$, b_w is defined as:

$$b_w = \arg\min_{b\in\mathbb{M}^w} \Big\{ \hat{I}(\boldsymbol{v}, b|\mathbf{X}, h^2\mathbf{I}) \Big\}, \tag{3.2.11}$$

where $\mathbb{M}^w$ is an interval defined by w *prominent* modes on $\boldsymbol{v}$. The value w represents the number of largest modes to consider as *prominent*. When $w = 2$, only the two largest modes are considered, with their location on $\boldsymbol{v}$ defining the interval within which b_w is defined. If $w = 4$, then the location of the four largest modes define the interval boundary for b_w, with the lowest associated location on $\boldsymbol{v}$ setting the lower bound and the highest associated modal location setting the upper bound. If w is greater than the number of modes identified within the estimated density, then w is truncated to the observed number of modes. If the estimated density is uni-modal then w is truncated to one and

the interval will consist of only one point on $\boldsymbol{v}$ representing the location of the mode, in which case we consider b_w to be undefined. When b_w is undefined over all projected density estimates, the solution reverts to the hyperplane solution along the initial projection vector. In situations when b_w is undefined for some of the projections, then the solution reverts to the last known defined b_w solution. Setting w excessively high, will result in considering lower valued probability modes and possibly lead to a solution that clusters all but a few observations into a single cluster. Figure 3.1 illustrates how the choice of w can affect the final location of the hyperplane with respect to relatively large α values.

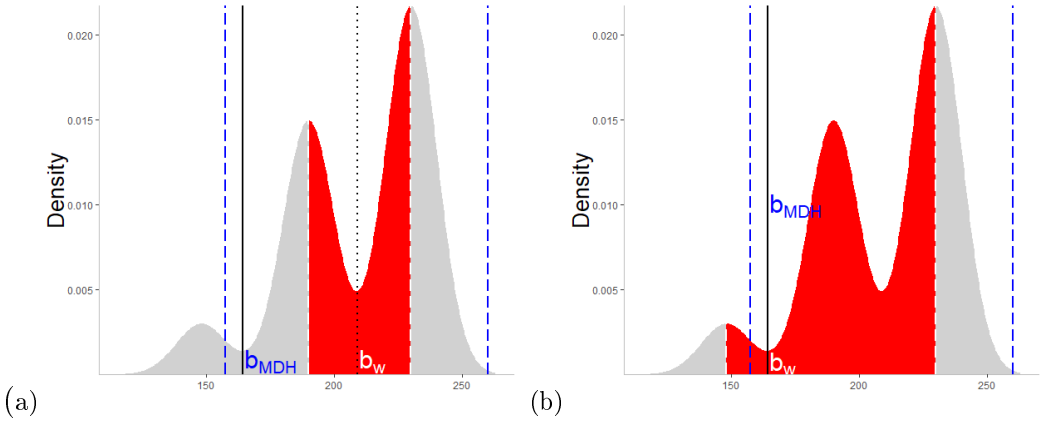


Figure 3.1: Example when setting $\alpha = 2$ and $w = 2$ with hyperplane solution b_w (a) and the setting where $w = 3$ with hyperplane solution $b_w = b_{MDH}$ (b). The blue lines represent the MDH maximum feasible region and the red shaded area represents the $\mathbb{M}^w$ interval.

Figure 3.1(a) illustrates the setting in which α is set relatively high (MDH feasible region indicated by blue lines). In this setting, MDH defines the optimal hyperplane at b_{MDH} (solid black line), clustering all but a relatively small number of objects to one group. However, implementing an additional constraint that restricts the final solution to a location between the two largest modes ($w = 2$) results in a final solution hyperplane at b_w. Increasing w to three results in an interval which contains the standard MDH's separator (Figure 3.1(b)) and thus both methods return the same hyperplane solution. The location of the hyperplane associated with $w = 2$ is arguably better, in the sense that the $\mathbb{M}^{w=2}$ interval contains the location of a hyperplane that results in a more balanced clustering solution. This simple example illustrates the motivation for applying an additional constraint upon the MDH solution. Additionally, bandwidth size should also be considered in concert with the selection of w since h influences the number of modes that an estimated density

will contain. The joint effect of w and bandwidth size are illustrated later on in Section 3.3.2. With the formulation of MDH defined, attention now turns to applying MDH within R.

3.3 Application in R

MDH clustering as defined within this text is applied using an augmented version of the `mdh` function from the `PPCI` R-package (Hofmeyr, 2018). The only required input to apply MDH within R is the dataset being clustered. There are several optional inputs a user can define when applying the standard `mdh` function from the `PPCI` package: the scale of the penalty term (α), an initial projection direction (defaulted to the first principal component) and a bandwidth to estimate the probability density functions. The original `mdh` function is adjusted to account for the constraint, dictated in Equation 3.2.11, which allows for the additional input for a number of *prominent* modes to consider when constructing the $\mathbb{M}^w$ interval. For the purpose of this text, focus is restricted to the choice of α, bandwidth and w parameters. Interested readers are directed to Hofmeyr (2018) for further details regarding the various inputs that can be specified within the `mdh` function.

MDH is applied to each of the data structure types displayed in Figure 2.2. Figure 3.2 illustrates how MDH assigns clusters for each of these distinct data structure types. As expected, MDH performs well when segmenting the linearly separable dataset (Figure 3.2(a), *Type A*) and does not correctly cluster the remaining non-linearly separable data types (Figure 3.2(b-c), *Type B-C*). However, projection pursuit enables a solution for *Type B* which is reasonable, resulting in relatively few errors compared to the other non-linearly separable datasets.

From each of the MDH solutions the projection vector ($\boldsymbol{v}$), allowable distance from mean (α), hyperplane location (b) and bandwidth (h) can be extracted. Transforming the data via $\boldsymbol{v}^\top \mathbf{X}$, estimating the density using h and plotting the hyperplane at b allows for a visual inspection of the final MDH solution (Figure 3.3). *Type A* and *Type B* datasets exhibit bi-modal distributions with a hyperplane solution that correctly identifies the minimum integrated density within the feasible region. *Type C* and *Type D* solutions also locate the minimum density within the feasible region. The estimated density for *Type C* contains more than two modes and as such it is not possible to locate a point on $\boldsymbol{v}$ that will successfully cluster each object to its true class. For MDH to accurately cluster densities which are multimodal, each class would have to be positioned entirely to one side of the hyperplane location on $\boldsymbol{v}$. MDH was unable to identify more than one mode for the *Type D* dataset. Ultimately there is no location on $\boldsymbol{v}$ that will result in successfully grouping all points for the *Type D* dataset.

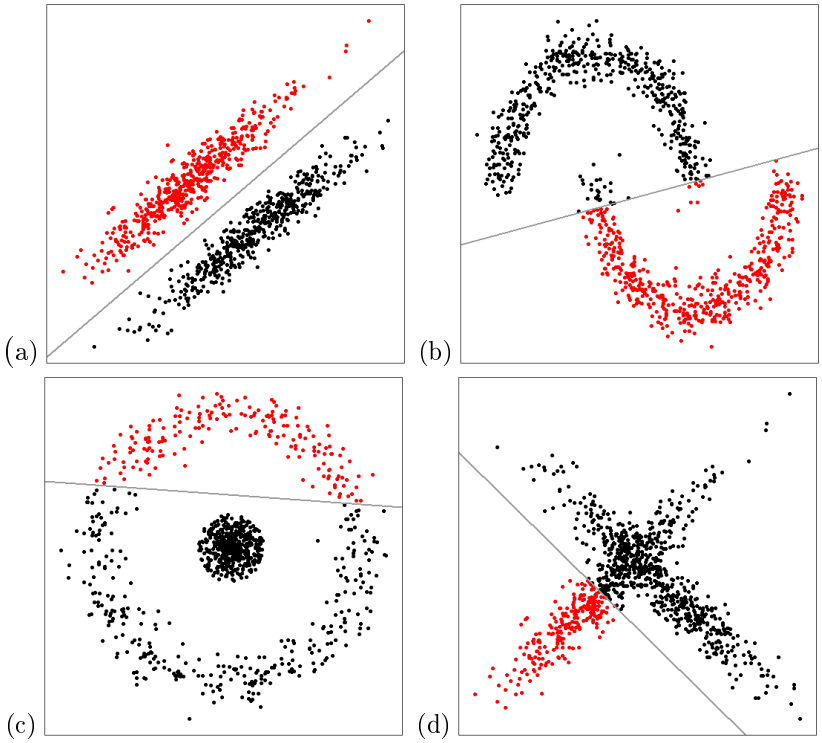


Figure 3.2: Minimum Density Hyperplane clustering of distinct (a-c) and overlapping (d) group structures.

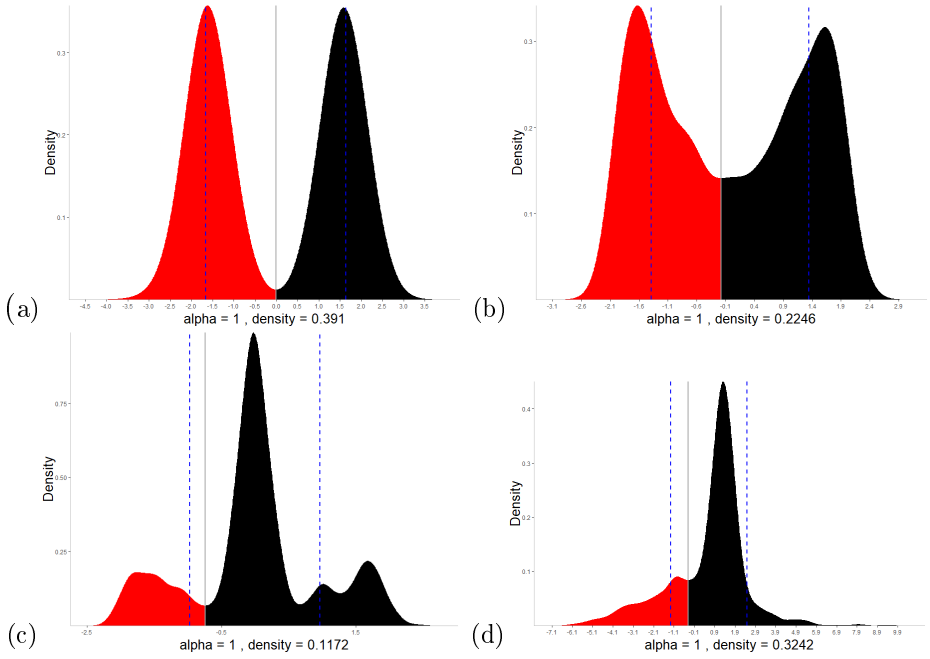


Figure 3.3: Univariate density estimates used for final results from clustering distinct (a-c) and overlapping (d) cluster structures.

3.3.1 Effect of bandwidth

As with all density-based methods, the choice of kernel bandwidth is crucial. Silverman (1986) recommends a bandwidth selection rule, $h = 0.9\hat{n}^{-1/5}\hat{\sigma}$, where 0.9 is a heuristic for univariate data, assumed to be multimodal and $\hat{\sigma}$ is the standard deviation of the data. MDH adopts this bandwidth selection rule but replaces the estimated standard deviation of a dataset with the square root of the variance explained within the first principal component. The bandwidth selection rule is defined as:

$$h = 0.9\hat{n}^{-1/5}\hat{\sigma}_{pc_1}, \tag{3.3.1}$$

where $\hat{\sigma}_{pc_1}$ is the estimated standard deviation of the data projected on the first principal component (Pavlidis *et al.*, 2016). This bandwidth will be referred to as the *heuristic* kernel bandwidth in this text. The multivariate version of the heuristic is defined as:

$$h^* = \left\{\frac{4}{2+d}\right\}^{-1/(4+d)} n^{-1/(4+d)}\hat{\sigma}_{pc_d}, \tag{3.3.2}$$

where d represents the number of dimensions and $\hat{\sigma}_{pc_d}$ is the average estimated standard deviation of the data projected onto d principal components. The h^* bandwidth will be referred to as the *full* kernel bandwidth for the remainder of this text. Bear in mind that given the definition in Equation 3.3.2, it cannot be said that the heuristic applies greater kernel smoothing compared to the full bandwidth. The relative difference is dependent on the dataset used for clustering and in some instances, one will apply relatively more smoothing than the other. Unless otherwise stated, the heuristic bandwidth is utilised when clustering a dataset. Applying MDH to *Type A* and *Type C* data using various bandwidths illustrates the influence that the bandwidth has on the final solution (Figure 3.4, 3.5 respectively).

Applying smaller bandwidth values when clustering *Type A* further increases the disparity between each of the classes' associated densities. Increasing the bandwidth results in merging both modes and a solution which poorly clusters the observations. While MDH should easily find the optimal separating plane for data which are linearly separable, choosing a bandwidth that is relatively large will produce a poor solution. For data which are non-linearly separable, as with *Type C*, the choice of bandwidth has little impact on the final solution. No level of smoothing can achieve a density upon which a linear separator will successfully assign class labels of this data type.

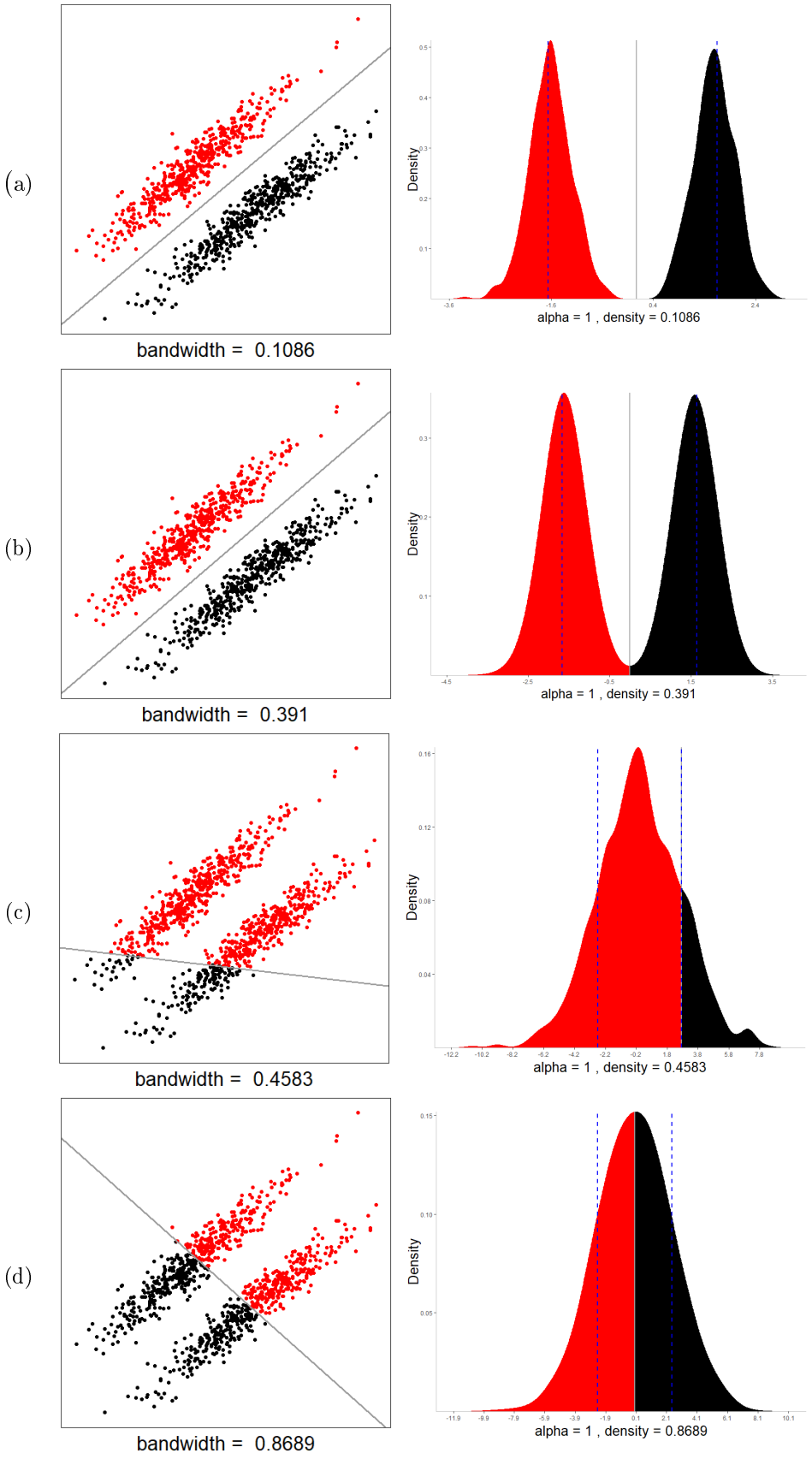


Figure 3.4: Effect of kernel bandwidth on MDH solution for distinct cluster *Type A*, using: a relatively small bandwidth(a), heuristic bandwidth (b), full bandwidth (c) and large bandwidth (d).

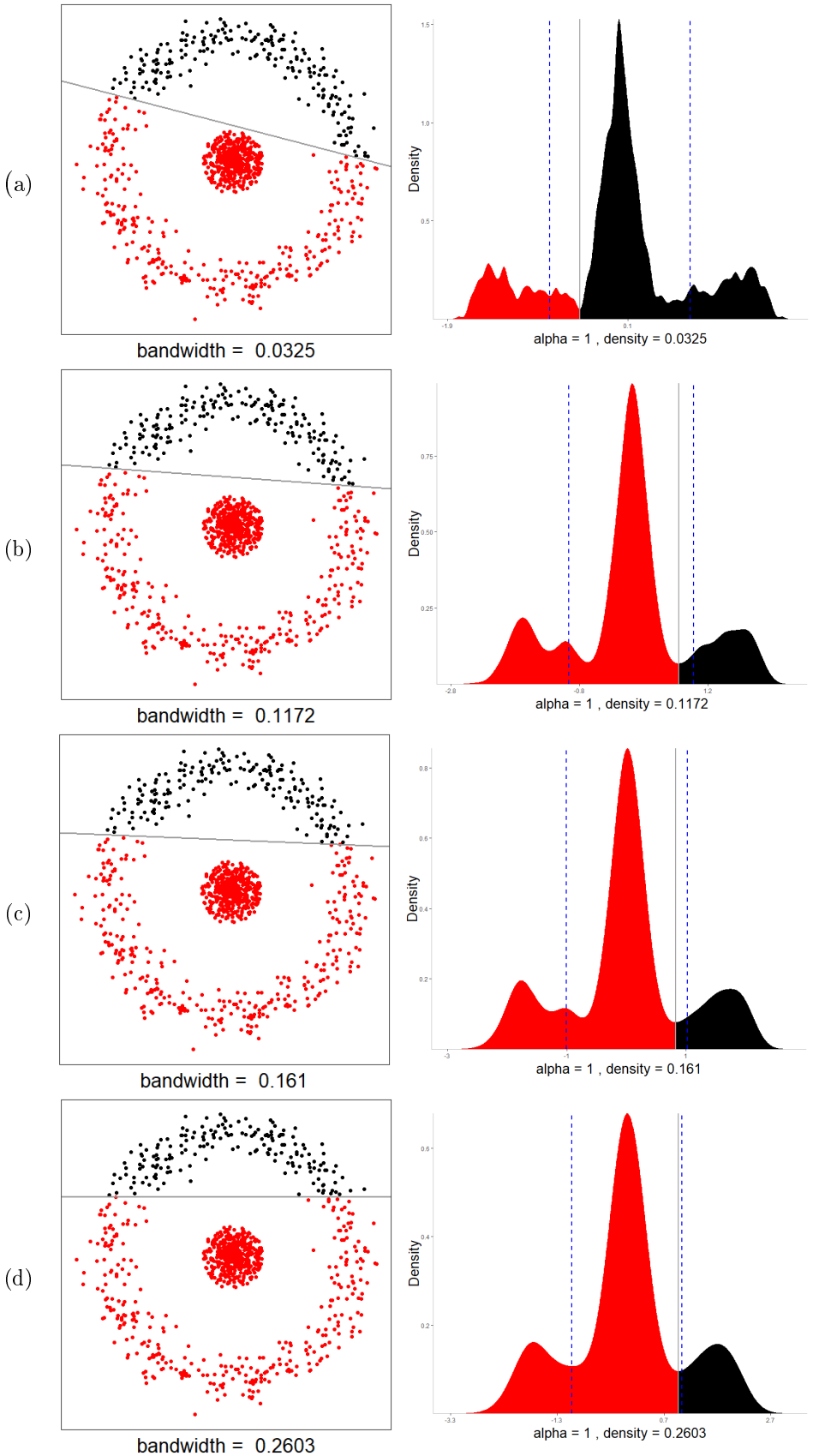


Figure 3.5: Effect of kernel bandwidth on MDH solution for distinct cluster *Type B*, using: a relatively small bandwidth(a), heuristic bandwidth (b), full bandwidth (c) and large bandwidth (d).

3.3.2 Constraints on hyperplane

As previously mentioned, the MDH solution plane is restricted to be within a given distance from the mean of the data. This reduces the possibility of assigning all but a few observations into one cluster. While the penalty does assist with reducing the possibility of returning this solution, it also serves another meaningful purpose. At each incremental α value, projection pursuit seeks an optimal transformation to obtain a low-density linear separating plane. Setting a larger range for α, increases the overall search region and iterations for projection pursuit. This increases the possibility of locating an optimal separating hyperplane.

The partial path to the final MDH solution for the *Type B* data is illustrated in Figure 3.6. At the first iteration, MDH utilises the first principal component as a projection vector and estimates the associated density function. Each subsequent projection's density is estimated by projecting the data onto the axis with maximum variance orthogonal to the projection vector. The points within each plot represent the projection of the data, where the y-axis represents the maximum variability orthogonal to the projection vector, associated with the x-axis (Pavlidis *et al.*, 2016). The red line within each plot indicates $\mathrm{H}(\boldsymbol{v}, b)$ and the constrained optimisation region is indicated as black dotted lines. As α incrementally increases, projection pursuit rotates the data. Iteration 16 represents the final, optimal hyperplane solution.

The solution associated with iteration 27 is rejected since it produces a hyperplane which passes through a point which is not a minimiser. This solution would have grouped all of the data into one cluster, as evident from the location of b relative to the scatter plot of all transformed data points. Based on the formulation of MDH, iteration 27's solution was rejected and MDH reverts back to the last known acceptable solution.

Previously, it was stated that w should be chosen in concert with the kernel bandwidth. Consider again the linearly separable *Type A* dataset. With a relatively large bandwidth and $w = 2$ (indicating that b must lay between the interval defined by the two largest modes), the result clusters all but a few objects to one cluster (Figure 3.7). While it is reasonable to expect a quality separating hyperplane resides between the two largest modes, this reasoning was diminished due to a poor choice of bandwidth. If the added constraint on the feasible region was not implemented, the hyperplane location on $\boldsymbol{v}$ would have been placed near -8, which would have resulted in one cluster containing all but a few observations.

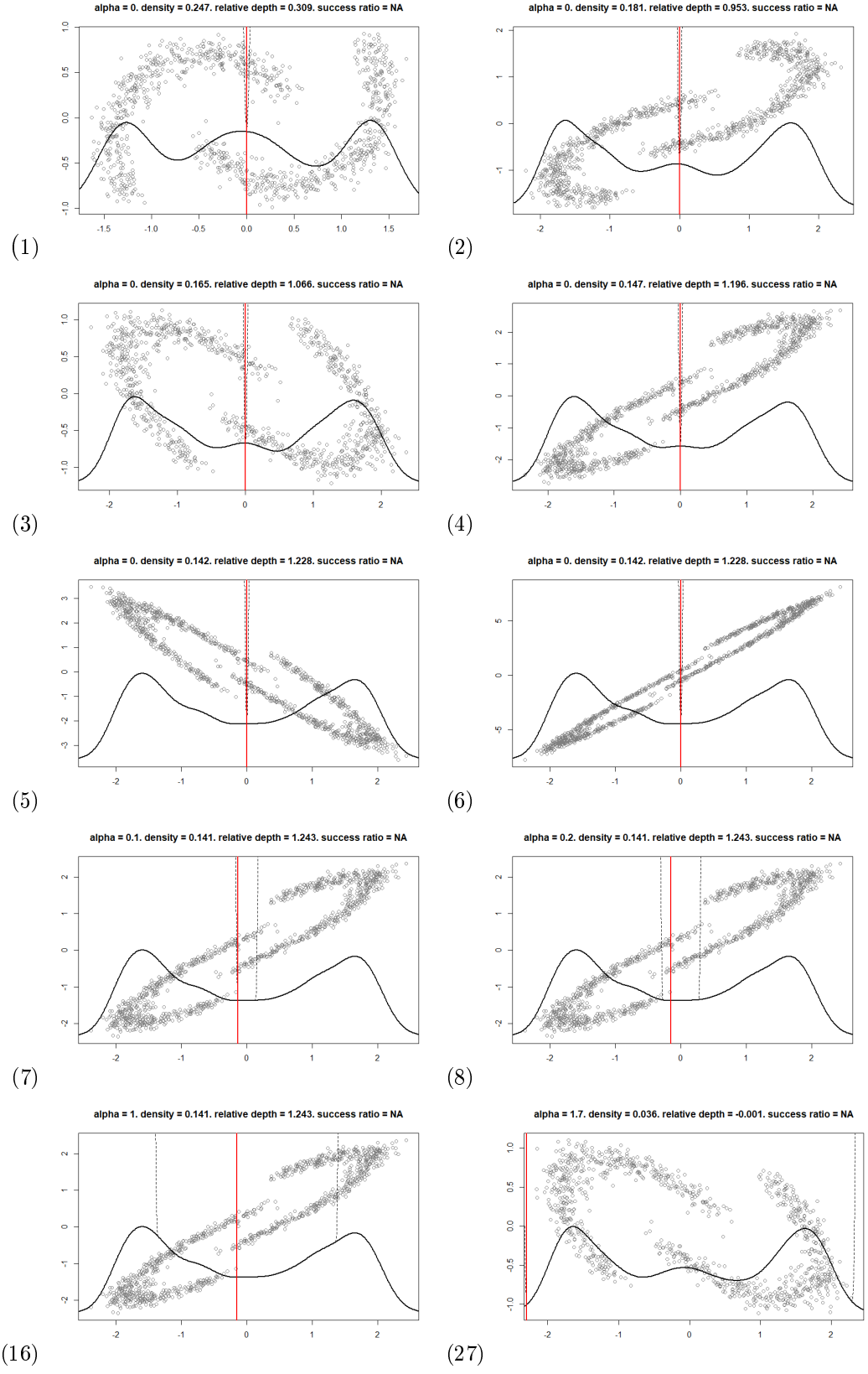


Figure 3.6: Illustration of Minimum Density Hyperplane estimation through different iterations utilising incremental α values. Each figure is accompanied by a number representing the overall iteration within the MDH solution.

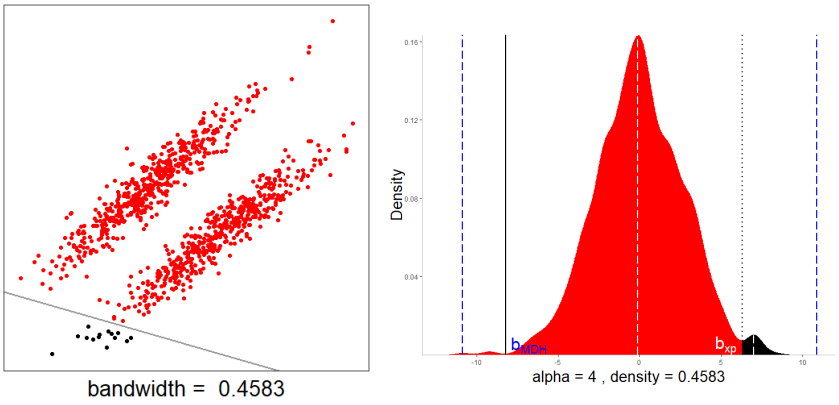


Figure 3.7: MDH solution of linearly separable data using a relatively large bandwidth and setting $w = 2$. The blue dashed lines represent the maximum feasible region, scaled be α and the white dashed lines represent the $\mathbb{M}^{w=2}$ interval

3.4 Summary

Minimum Density Hyperplane clustering was presented and details regarding how the algorithm solves the clustering problem were discussed. MDH seeks the minimum integrated density of an estimated probability distribution function. One of the challenges involved with locating a low-density hyperplane is to avoid solutions which groups all but a few observations to one cluster. Restricting the hyperplane distance to the mean of the data greatly reduces the likelihood of obtaining such a solution. An additional constraint was presented, whereby the final position of the hyperplane solution must reside within an interval bounded by the smallest and largest of the w *prominent* mode locations on $\boldsymbol{v}$. With this newly presented constraint, α can be set high, increasing the overall search area and projection pursuit iterations. This increases the possibility of locating an optimal separating hyperplane without the consequence of an insignificant solution. While this additional constraint guards against a solution which places all but a few objects into one cluster, it is still susceptible to the choice of bandwidth size.

As with all clustering methods the choice of bandwidth is not trivial. Clustering the linearly separable data, *Type A* with large h values produced poor solutions while relatively smaller values resulted in correctly identifying the underlying class structure. Regardless of the size of the kernel smoother, when data are non-linearly separable MDH cannot learn the true underlying cluster structure. Herein lies a limitation of an MDH solution, the linear decision boundary. Since MDH learns a linear decision boundary, it is incapable of segmenting complex cluster structures which are non-linearly separable. The following chapter details our approach to improve upon hyperplane solutions.

Chapter 4

Improving Hyperplane Solutions

4.1 Introduction

The main limitation of an MDH is that it defines clusters based on a linear decision boundary. Thus, MDH performs poorly when the clusters in the data are not linearly separable. One method to obtain a non-linear hyperplane involves mapping the dataset into a high-dimensional feature space using Kernel Principal Component Analysis (Schölkopf *et al.*, 1998, KPCA). MDH is then applied to the embedded data whereby the low-density separator in the feature space corresponds to a non-linear hyperplane in the input space (Yates and Pavlidis, 2016). However, KPCA does not scale well to larger datasets due to its quadratic space and time complexity. This section presents a novel approach to improve hyperplane solutions in an efficient manner, by reassigning points in a neighbourhood around a separator using a more flexible approach, namely Mean Shift (Cheng, 1995).

The procedure begins by collecting observations within a region around the hyperplane. The region will be referred to as the *Gamma region*. The points in the dataset which fall in the region is defined as:

$$\boldsymbol{\Gamma}_L = \mathbf{X}^+_{\boldsymbol{v},b,L} \cup \mathbf{X}^-_{\boldsymbol{v},b,L} \tag{4.1.1}$$

$$\mathbf{X}^-_{\boldsymbol{v},b,L} := \Big\{ \mathbf{x} \in \mathbf{X} \Big| b - L^{\pm} \leq \boldsymbol{v}^\top \mathbf{x} < b \Big\}, \tag{4.1.2}$$

$$\mathbf{X}^+_{\boldsymbol{v},b,L} := \Big\{ \mathbf{x} \in \mathbf{X} \Big| b + L^{\pm} \geq \boldsymbol{v}^\top \mathbf{x} \geq b \Big\}, \tag{4.1.3}$$

where $L^{\pm}$ is defined as:

$$L^{\pm} = \Big(\max\{\boldsymbol{v}^\top \mathbf{x}\} - \min\{\boldsymbol{v}^\top \mathbf{x}\} \Big) \times L/2, \tag{4.1.4}$$

where L represents the proportion of the range of the data to consider for reassignment. As an example, consider a setting in which the maximum value

of $\{\boldsymbol{v}^\top \mathbf{x}\}$ is 100 and the minimum is 0. Furthermore, consider that the hyperplane is located at 75. Now if $L = 0.10$, then the Gamma region will consist of those observations that are associated with projected values between 70 and 80. It is important to note that 10 per cent of the data's range around the hyperplane does not equate to 10 per cent of data. In fact, there are cases in which the Gamma region will contain no observations. This occurs when a dataset contains compact clusters that exhibit high inter-cluster separation and results in a hyperplane located within a sparse region. An example of such a scenario is illustrated in the appendix (Figure B.3).

Applying a more flexible density-based approach to the Gamma region allows the final solution to be non-linear. Mean Shift clustering is a sensible technique to apply to points in a neighbourhood around a hyperplane solution, since it assigns data to clusters in a flexible way without the constraint of a linear decision boundary (Cheng, 1995). Mean Shift is applied to each observation and reassigns them according to the location of their associated modal point relative to the hyperplane. Since this approach only applies MS to a subset of the data, it dramatically reduces the time and space complexity of applying MS to the entire dataset, allowing application within larger datasets. Applying MS to points in a candidate region around the MDH solution hyperplane will be denoted as $\text{MDH}_{\boldsymbol{\Gamma}_{MS}}$.

To mitigate the computational cost of MS, one can use a single step gradient approach. This is achieved by calculating the derivative of the probability density function evaluated at each object and then assigning a label to each based on the gradient direction relative to the hyperplane. If the estimated slope points in a direction towards the hyperplane then it is likely that the object's gradient ascent trajectory will converge beyond the hyperplane and thus it should be reassigned. Conversely, if the gradient points in a direction away from the hyperplane then it is likely that the object will converge to a mode within the cluster it was assigned by MDH and is thus not reassigned. Essentially, this is the single step heuristic approach of MS. This procedure is denoted as $\text{MDH}_{\boldsymbol{\Gamma}_{H}}$. The motivation of this heuristic is illustrated in greater detail within Section 4.3.2.

We have defined the Gamma region to be the collection of observations around a low-density separator and presented two flexible techniques to improve hyperplane solutions. This chapter will evaluate the performance of these MDH enhancements. The remaining chapter is organised as follows: First, Mean Shift is formulated and applied to the set of distinct cluster types. This is followed by the application of $\text{MDH}_{\boldsymbol{\Gamma}_{MS}}$ using a variety of Gamma regions. Then its heuristic counterpart is detailed before applying it to the set of distinct cluster data types. Before concluding this chapter the performance of MDH, $\text{MDH}_{\boldsymbol{\Gamma}_{MS}}$ and $\text{MDH}_{\boldsymbol{\Gamma}_{H}}$ are evaluated across a range of benchmark datasets. Additionally, interested parties can interactively cluster simulated data with various bandwidths and Gamma region sizes, using MDH, $\text{MDH}_{\boldsymbol{\Gamma}_{MS}}$ and $\text{MDH}_{\boldsymbol{\Gamma}_{H}}$ online at *https://jacobbradleykenyon.shinyapps.io/ClusterSim*.

4.2 Mean Shift Clustering

Mean Shift clustering is a simple and flexible procedure that assigns objects to clusters in a non-linear way. In this section, MS is formulated and then applied to the distinct cluster data types.

Mean Shift is known as a gradient ascent technique which can cluster complex datasets and does not make any assumptions about the true underlying number of clusters (Cheng, 1995). It defines a cluster as a collection of observations within an attraction basin. Recall that an attraction basin is the region containing a group of objects whose gradient ascent trajectory terminates at a single, shared mode. The MS procedure calculates a locally weighted average around each observation, weighted by their kernels, and then shifts objects to their local mean. This process is repeated until all observations have converged to an estimated mode (Cheng, 1995).

Consider again the finite dataset, $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{n} \subset \mathbb{R}^d$, from an unknown probability density function $p(\mathbf{x})$. The kernel density estimator is defined as:

$$\hat{p}(\mathbf{x}|\mathbf{X}, h^2\mathbf{I} = \mathbf{H}) = \frac{1}{n}\sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_i), \tag{4.2.1}$$

where $K(\mathbf{x}) = (2\pi)^{-d/2} exp(-\frac{1}{2}\mathbf{x}^\top\mathbf{x})$ is the symmetric multivariate Gaussian probability density function and $K_{\mathbf{H}}(\mathbf{x}) = |\mathbf{H}|^{-1/2}K(\mathbf{H}^{-1/2}\mathbf{x})$. For the purpose of applying MS to a collection of objects around the hyperplane, a bandwidth equivalent to that which was used to derive the MDH hyperplane will be applied, viz. $\mathbf{H} = h^2\mathbf{I}$. From the kernel density estimator, the slope of a function evaluated at each element is defined as:

$$\nabla\hat{p}(\mathbf{x}|\mathbf{X}, h^2\mathbf{I} = \mathbf{H}) = \frac{1}{nh}\sum_{i=1}^{n} K_{\mathbf{H}}^{'}(\mathbf{x} - \mathbf{x}_i)(\mathbf{x} - \mathbf{x}_i). \tag{4.2.2}$$

Setting this equal to 0, we have

$$\sum_{i=1}^{n} K_{\mathbf{H}}^{'}(\mathbf{x} - \mathbf{x}_i)\mathbf{x} = \sum_{i=1}^{n} K_{\mathbf{H}}^{'}(\mathbf{x} - \mathbf{x}_i)\mathbf{x}_i, \tag{4.2.3}$$

with

$$\mathbf{x} = \frac{\sum_{i=1}^{n} K_{\mathbf{H}}^{'}(\mathbf{x} - \mathbf{x}_i)\mathbf{x}_i}{\sum_{i=1}^{n} K_{\mathbf{H}}^{'}(\mathbf{x} - \mathbf{x}_i)}, \tag{4.2.4}$$

and assuming $g(\mathbf{x}) = -K_{\mathbf{H}}^{'}(\mathbf{x})$:

$$m(\mathbf{x}) = \frac{\sum_{i=1}^{n} g(\mathbf{x} - \mathbf{x}_i)\mathbf{x}_i}{\sum_{i=1}^{n} g(\mathbf{x} - \mathbf{x}_i)} - \mathbf{x}. \tag{4.2.5}$$

The quantity $m(\mathbf{x})$ is known as the mean shift vector. A summary of the Mean Shift procedure is as follows; at each point $\mathbf{x}_i$, compute the mean shift vector $m(\mathbf{x}_i^\top)$, update $\mathbf{x}_i^\top$ with the kernel weighted average at time t ($m(\mathbf{x}_i^\top) = \mathbf{x}_i^{t+1}$) and repeat until convergence. With the formulation defined according to Cheng (1995) and Duong *et al.* (2007), attention now turns applying MS to the distinct cluster type datasets.

The Mean Shift algorithm as described above is applied using the `kms` function from the `ks` R-package (Duong, 2018). A benefit of this package is that it allows one to easily extract the gradient ascent path of each data point. Figure 4.1 illustrates the final cluster assignment for each of the different data types. For comparability with MDH, the minimum cluster size was set to 245 so that the final solution would result in a binary partition. Also, a bandwidth exact to that which was used for the MDH solution was utilised to cluster the data (heuristic bandwidth detailed in Equation 3.3.1). If a cluster contained less than the minimum required observations, then it was merged using a centroid linkage hierarchical approach based on Euclidean distances.

For the linearly separable data, *Type A*, MS incorrectly assigned 3 observations to the black cluster. It was found that this occurred since those observations are represented by a local mode whose nearest estimated modal neighbour was represented by the black cluster and as such these points were assigned to the black cluster (Figure B.1). For the *Type B* data, MS successfully assigned all objects to their true cluster. MS failed to identify the true clusters for the *Type C* dataset and instead divided the ring of points into multiple smaller clusters. Then those clusters containing fewer than 245 observations were merged with the nearest cluster, based on the distances between cluster modes (Figure B.1). Once a cluster is merged with another, the cluster centroid (mode) is updated. This process repeats until all clusters contain at least 245 objects. Thus, the red points of the outer ring had estimated modes which were closest to the centre sphere of points. Without this restriction (Figure B.2(a)), the outer ring elements would not have been assigned to the cluster representing the inner sphere of observations. Results from MS clustering *Type D* is similar to MDH and reiterates the difficulty of clustering overlapping data structures using density-based approaches. The full unrestricted MS solution (no required cluster size) is illustrated within the appendix (Figure B.2(b)).

The clustering performance of MDH and MS on the different data types is evaluated using the Success Ratio (SR) and average silhouette coefficient (Table 4.1). MDH successfully clustered *Type A* (Figure 3.1(a), thus SR=1. MS misclassified a few points with regard to clustering *Type A* and results in SR=0.994. The silhouette coefficient reiterates these findings with MDH having a larger value than MS, indicating a better quality clustering solution. However, besides the results associated with the *Type A* dataset, the silhouette coefficient produced misleading values which did not comply with the actual results. This illustrates the limitation associated with internal cluster validation

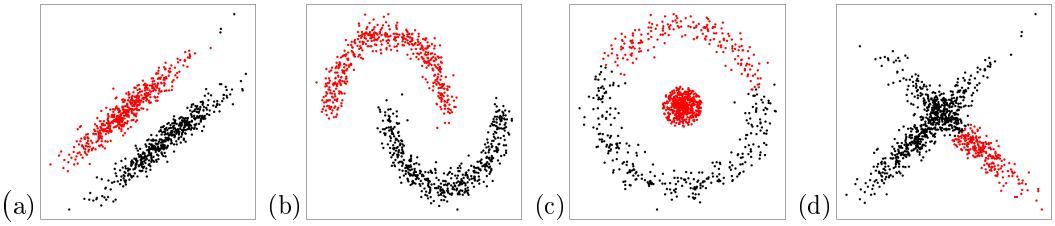


Figure 4.1: Mean Shift clustering of distinct (a-c) and non-distinct (d) grouping. Minimum cluster size was set to 245 observations.

metrics. Recall that for *Type B* data, MS correctly clustered all observations while MDH clustered a majority of the points accurately, with SR=1 and 0.929 respectively. MDH was unable to distinguish at least one cluster containing the majority of its true class for the complex and overlapping cluster structures (*Type C* and *Type D*), resulting in an SR=0 for each instance. MS was able to distinguish at least one cluster containing the majority of its true class for *Type C*, with SR=0.606. This is due to MS assigning fewer outer ring points to the centre sphere of observations. The MDH solution from clustering the *Type B* dataset can be improved by applying MS to a Gamma region that is large enough to reassign all the incorrectly MDH clustered points. This will improve the overall quality of clustering and thus increase the Success Ratio from 0.929 to 1.

Table 4.1: Comparison of MDH and MS solutions.

	Type	Success Ratio	Silhouette Coefficient
Minimum Density Hyperplane	A	**1.000**	**0.376**
	B	0.929	**0.470**
	C	0	**0.428**
	D	0	0.391
Mean Shift*	A	0.994	0.375
	B	**1.000**	0.451
	C	**0.606**	0.410
	D	0	**0.411**

MDH solutions were obtained using the heuristic bandwidth scalar, h, MS utilised $h^2\mathbf{I}$.
Bold indicates *best* clustered solution per validation metric.
* MS results are based on the restricted solution, minimum cluster size of 245 observations.

4.3 Non-linear Extensions

Mean Shift has been formulated and proposed as a method to improve a hyperplane solution. In this subsection, MS is applied to a collection of points around the MDH hyperplane solutions obtained from clustering the distinct data types (Figure 3.1).

4.3.1 Mean Shift Reassignment

The *Type B* MDH solution is now re-evaluated around the hyperplane. A candidate region containing L=0.10, 0.20, 0.25, 0.30 and 0.35 are considered for reassignment. A diagonal kernel bandwidth matrix equal to the heuristic used to obtain the MDH solution, $\mathbf{H} = h^2\mathbf{I}$, was utilised during the MS procedure. Figure 4.2 illustrates the final reassigned values for the *Type B* dataset. The final cluster label is indicated as either black or red with the Gamma region highlighted in green. Increasing L from 0.10 to 0.35 increases the region considered for reassignment and MS correctly reassigns all points for each setting. $\text{MDH}_{\mathbf{\Gamma}_{MS(0.3)}}$ covers an area which contained all of the original MDH clustering errors and results in a Success Ratio equal to one. This was expected since MS was found to correctly cluster the *Type B* dataset. Applying MS to a candidate region around the MDH hyperplane amounts to replacing the MDH solution of said region with MS. Thus, increasing L results in replacing more MDH cluster labels with the MS solution.

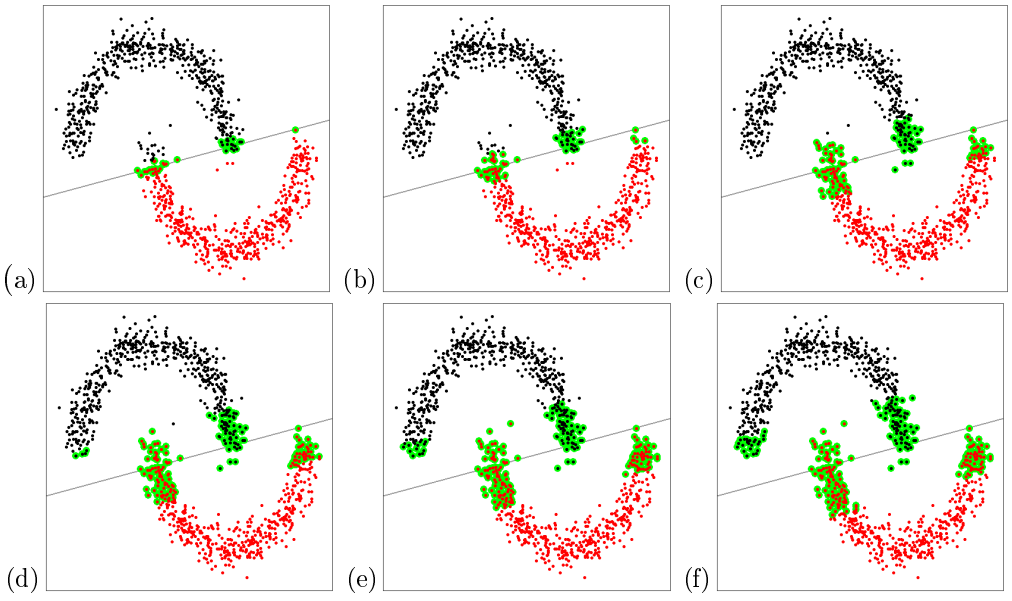


Figure 4.2: Mean Shift reassignment of distinct cluster *Type B* MDH solution over: $\mathbf{\Gamma}_{0.05}$ region (a), $\mathbf{\Gamma}_{0.10}$ region (b), $\mathbf{\Gamma}_{0.20}$ region (c), $\mathbf{\Gamma}_{0.25}$ region (d), $\mathbf{\Gamma}_{0.30}$ region (e),$\mathbf{\Gamma}_{0.35}$ region (f).

One might consider setting a relatively large value for L but there is an obvious trade-off between flexibility and computation when adjusting the Gamma region size. Setting a relatively large value of L provides greater flexibility but at the cost of increasing the computation required to reassign all Gamma region points. By applying the single step heuristic approach as opposed to MS greatly reduces the trade-off between flexibility and computation. The following section reviews the single step gradient heuristic approach. Results from MS clustering in the case of the other data types are located in the appendix (Figures B.3, B.4 and B.5) which ingeminate the process of applying MS to the different MDH solutions.

4.3.2 Single Step Gradient Reassignment

One of the main constraints associated with reassigning observations using MS is that a point is only assigned to a cluster after it has converged to an estimated mode. This requires an increasing amount of iterations for larger datasets. Applying the single step gradient heuristic greatly reduces the computation of reassigning each observation within the Gamma region by removing the convergence criterion. This single step heuristic reassigns observations based on the estimated slope of a probability density function evaluated at each object. If an initial gradient points toward the hyperplane, then it is likely that the subsequent ascent trajectory will converge beyond the hyperplane, indicating the object should be reassigned to a different cluster. Consider that in gradient ascent, $\mathbf{x}^{t+1} = \mathbf{x}^t + c\nabla\hat{p}(\mathbf{x}^t)$, where c represents some small constant value. The projection onto $\boldsymbol{v}$ at step $t+1$ will be greater than that at step t, if and only if $\boldsymbol{v}^\top\nabla\hat{p}(\mathbf{x}^t)$ is positive and thus the observation is assigned to the cluster above the hyperplane. The heuristic reassignment rule is defined as:

$$\boldsymbol{v}^\top\nabla\hat{p}(\boldsymbol{\gamma}) > 0, \textit{ assign to cluster above hyperplane}, \tag{4.3.1}$$

$$\boldsymbol{v}^\top\nabla\hat{p}(\boldsymbol{\gamma}) \leq 0, \textit{ assign to cluster below hyperplane}, \tag{4.3.2}$$

where $\boldsymbol{\gamma} \in \boldsymbol{\Gamma}_L$.

As with all heuristics, this approach is not guaranteed to locate the true MS solution. For instance, consider a positive $\boldsymbol{v}^\top\nabla\hat{p}(\boldsymbol{\gamma})$, while it is likely that the gradient ascent path converges above the hyperplane it is found that the observation is actually located within a local density which has an associated mode that is below the hyperplane. The likelihood of these occurrences increases with larger L values. However, these instances may be deemed acceptable when considering the substantial reduction in computation required compared to MS. With that, the choice between applying MS or its heuristic is dependent on the size of data being clustered and whether the inherent errors produced by the heuristic are deemed acceptable.

$\text{MDH}_{\boldsymbol{\Gamma}_H}$ was applied to a variety of Gamma region settings around the MDH solution hyperplane obtained from the *Type B* dataset (Figure 4.3).

With relatively small L values, $\text{MDH}_{\mathbf{\Gamma}_H}$ accurately reassigns each observation. When $L = 0.20$ one of the objects are reassigned correctly. Applying the heuristic to a Gamma region with $L = 0.35$ results in a greater number of errors. Overall, the number of reassignment errors increases as the Gamma region expands. Similar illustrations regarding the other data types can be evaluated within Appendix B.2.1.

To better understand why this heuristic is appropriate and when errors can occur, a subset of observations' gradient ascent trajectories are plotted atop the final $\text{MDH}_{\mathbf{\Gamma}_{H(0.35)}}$ solution (Figure 4.4).

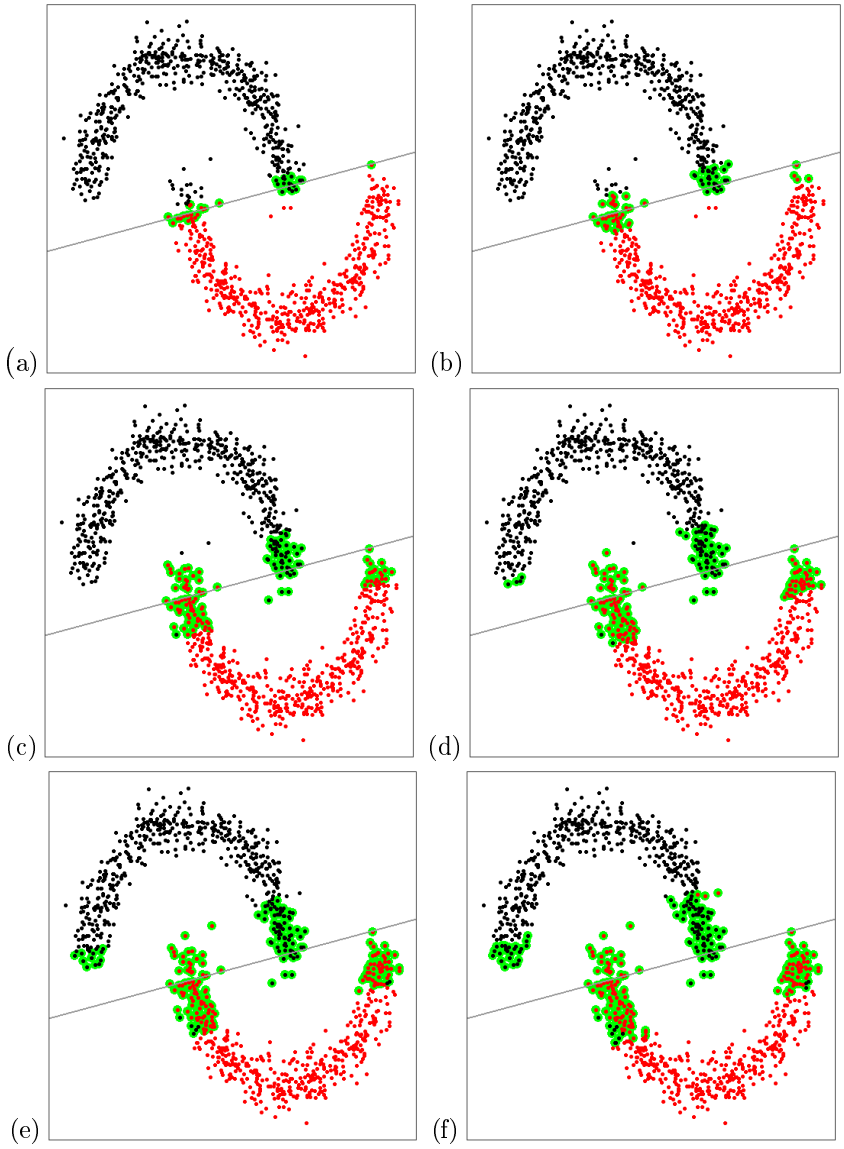


Figure 4.3: Heuristic reassignment of distinct cluster *Type B* MDH solution over: $\mathbf{\Gamma}_{0.05}$ region (a), $\mathbf{\Gamma}_{0.10}$ region (b), $\mathbf{\Gamma}_{0.20}$ region (c), $\mathbf{\Gamma}_{0.25}$ region (d), $\mathbf{\Gamma}_{0.30}$ region (e) and $\mathbf{\Gamma}_{0.35}$ region (f).

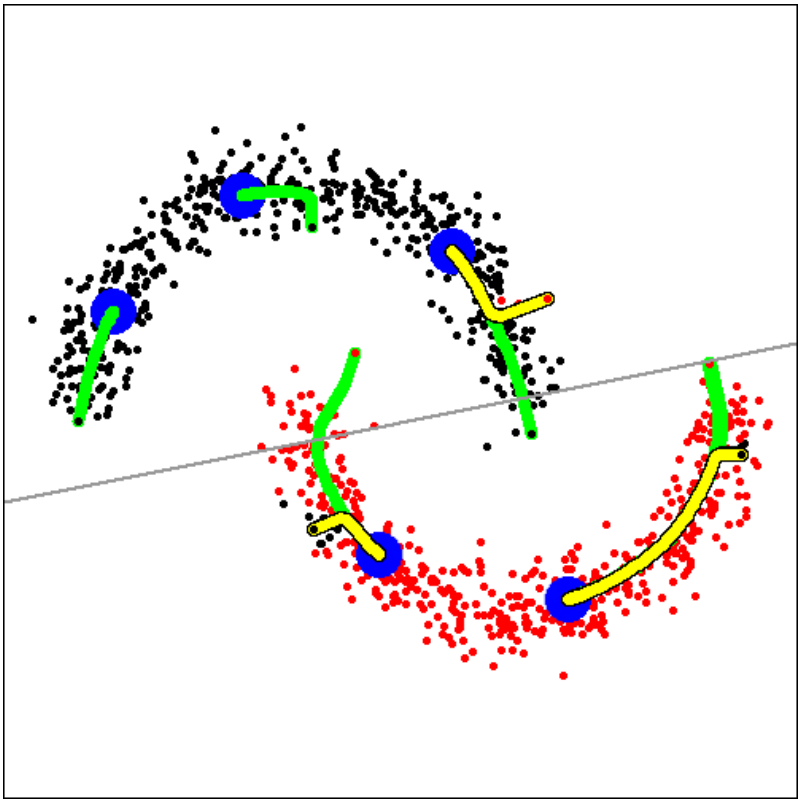


Figure 4.4: Gradient heuristic *Type B* solution using $\mathbf{\Gamma_{0.35}}$. Large blue points represent the Mean Shift estimated modes, green lines indicate gradient ascent trajectories with yellow indicating those which initially move towards the hyperplane but do not converge beyond the hyperplane.

It can be seen that those observations which were wrongly clustered had gradient ascent trajectories which initially moved towards but not beyond the hyperplane (indicated by yellow paths in Figure 4.4). Most of the gradient ascent trajectories did not change direction in relation to the hyperplane and were correctly reassigned. This solidifies the validity of this heuristic approach. However, the inherent limitation associated with applying this heuristic is exacerbated given larger L values. A similar figure illustrates the results from applying $\text{MDH}_{\Gamma_{H(0.35)}}$ to the *Type D* dataset and can be found in the appendix (Figure B.9). To test the validity of these proposed enhancements to the hyperplane solution, a series of benchmark tests is undertaken on real-world datasets obtained from the UCI Machine Learning Repository (Dheeru and Karra Taniskidou, 2017).

4.4 Benchmark Tests

$\text{MDH}_{\mathbf{\Gamma}_{MS}}$ and $\text{MDH}_{\mathbf{\Gamma}_{H}}$ were able to successfully enhance the linear hyperplane solution for the bi-variate, *Type B* dataset. In this section, MDH and its enhancements are applied to a variety of benchmark datasets. For each dataset, only a subset of observations were utilised so that each dataset contained only two classes. The performance of each method will be measured using the Success Ratio. While the Success Ratio is capable of handling datasets with more than two classes, this study evaluates only a subset of each dataset to reduce computation. Information regarding groups that were utilised for the study are found in the appendix (Table B.1).

To begin, each dataset was clustered using MDH with a maximum α set to 1.5. Each analysis was applied to two different MDH solutions; one which produced the standard MDH solution ($w = 0$) and another which required the final hyperplane to reside between the two largest modes ($w = 2$). In cases when the hyperplane's location across all projected densities does not lie within the $\mathbb{M}^2$ interval, the solution reverts to the hyperplane along the initial projection. The heuristic (h), full (h^*) and an experimental bandwidth (h_{xp}) were applied to each dataset to investigate the impact of the bandwidth size (Table 4.2). The experimental bandwidth is a procedure that searches for an ideal bandwidth over a range of values. At each value of h, the minimum density within the feasible region of the initial projection is located. Then the number of observations within a set Gamma region (e.g. $L = 0.25$) around the hyperplane is tabulated. The value of h associated with the fewest number of observations within the Gamma region is considered the ideal bandwidth (h_{xp}) for application in MDH and the enhancement procedures. Further details regarding h_{xp} can be found in Appendix B.3. The Success Ratios from each setting are illustrated in Tables 4.3 to 4.8. In each table, the best method is indicated in bold, with ties awarded to the method requiring the least amount of computation. Green and red arrows indicate if the MDH solution was improved or degraded. A discussion of the results contained in the tables follows in Sections 4.4.1 to 4.4.3.

Table 4.2: Details of benchmark datasets.

Dataset	n	d	h	h^*	h_{xp}
Banknote	1372	4	1.499	1.310	0.999
Seeds	70	7	1.016	0.385	0.677
Wine	66	13	138.663	21.105	55.866
Votes	435	16	0.362	0.355	6.026
Breast Cancer	569	30	168.580	20.732	116.133
Synthetic Control	100	60	9.687	4.113	10.995

Details of benchmark datasets: size(n), dimensionality(d), calculated heuristic (h), full (h^*) and *optimised* (h_{xp}) bandwidths.

Table 4.3: Benchmark datasets' Success Ratios using $w = 0$ and h.

	MDH	$\text{MDH}_{\mathbf{\Gamma}_{MS}}$			$\text{MDH}_{\mathbf{\Gamma}_{H}}$		
		$\mathbf{\Gamma}_{0.10}$	$\mathbf{\Gamma}_{0.20}$	$\mathbf{\Gamma}_{0.30}$	$\mathbf{\Gamma}_{0.10}$	$\mathbf{\Gamma}_{0.20}$	$\mathbf{\Gamma}_{0.30}$
Banknote	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000
Seeds	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000
Wine	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000
Votes	0.695	0.695	0.695	0.695	**0.711** ↑	0.698 ↑	0.706 ↑
Breast Cancer	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000
Synthetic Control	**0.860**	0.860	0.860	0.860	0.860	0.860	0.860

Table 4.4: Benchmark datasets' Success Ratios using $w = 0$ and h^*.

	MDH	$\text{MDH}_{\mathbf{\Gamma}_{MS}}$			$\text{MDH}_{\mathbf{\Gamma}_{H}}$		
		$\mathbf{\Gamma}_{0.10}$	$\mathbf{\Gamma}_{0.20}$	$\mathbf{\Gamma}_{0.30}$	$\mathbf{\Gamma}_{0.10}$	$\mathbf{\Gamma}_{0.20}$	$\mathbf{\Gamma}_{0.30}$
Banknote	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000
Seeds	0.861	0.833 ↓	0.833 ↓	0.833 ↓	**0.886** ↑	0.857 ↓	0.811 ↓
Wine	0.694	0.694	0.694	0.694	**0.750** ↑	0.611 ↓	0.568 ↓
Votes	0.714	0.714	0.714	0.714	0.719 ↑	**0.730** ↑	0.722 ↑
Breast Cancer	**0.610**	0.610	0.610	0.610	0.584 ↓	0.599 ↓	0.464 ↓
Synthetic Control	**0.608**	0.608	0.608	0.608	0.600 ↓	0.492 ↓	0.435 ↓

Table 4.5: Benchmark datasets' Success Ratios using $w = 0$ and h_{xp}.

	MDH	$\text{MDH}_{\mathbf{\Gamma}_{MS}}$			$\text{MDH}_{\mathbf{\Gamma}_{H}}$		
		$\mathbf{\Gamma}_{0.10}$	$\mathbf{\Gamma}_{0.20}$	$\mathbf{\Gamma}_{0.30}$	$\mathbf{\Gamma}_{0.10}$	$\mathbf{\Gamma}_{0.20}$	$\mathbf{\Gamma}_{0.30}$
Banknote	0.506	0.507 ↑	0.507 ↑	0.507 ↑	**0.507** ↑	0.000 ↓	0.000 ↓
Seeds	0.892	0.917 ↑	0.943 ↑	0.943 ↑	0.917 ↑	**0.943** ↑	0.833 ↓
Wine	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000
Votes	0.718	**0.731** ↑	0.706 ↓	0.695 ↓	0.726 ↑	0.674 ↓	0.603 ↓
Breast Cancer	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000
Synthetic Control	**0.640**	0.640	0.640	0.640	0.640	0.640	0.608 ↓

Table 4.6: Benchmark datasets' Success Ratios using $w = 2$ and h.

	MDH	$\text{MDH}_{\mathbf{\Gamma}_{MS}}$			$\text{MDH}_{\mathbf{\Gamma}_{H}}$		
		$\mathbf{\Gamma}_{0.10}$	$\mathbf{\Gamma}_{0.20}$	$\mathbf{\Gamma}_{0.30}$	$\mathbf{\Gamma}_{0.10}$	$\mathbf{\Gamma}_{0.20}$	$\mathbf{\Gamma}_{0.30}$
Banknote	0.422	0.459 ↑	0.486 ↑	0.501 ↑	0.460 ↑	**0.518** ↑	0.510 ↑
Seeds	**0.892**	0.861 ↓	0.886 ↓	0.886 ↓	0.861 ↓	0.886 ↓	0.778 ↓
Wine	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000
Votes	0.695	0.695	0.695	0.695	**0.711** ↑	0.698 ↑	0.706 ↑
Breast Cancer	**0.763**	0.645 ↓	0.528 ↓	0.000 ↓	0.645 ↓	0.397 ↓	0.309 ↓
Synthetic Control	**0.482**	0.482	0.482	0.482	0.000 ↓	0.338 ↓	0.342 ↓

Table 4.7: Benchmark datasets' Success Ratios using $w = 2$ and h^*.

	MDH	$\text{MDH}_{\mathbf{\Gamma}_{MS}}$			$\text{MDH}_{\mathbf{\Gamma}_{H}}$		
		$\mathbf{\Gamma}_{0.10}$	$\mathbf{\Gamma}_{0.20}$	$\mathbf{\Gamma}_{0.30}$	$\mathbf{\Gamma}_{0.10}$	$\mathbf{\Gamma}_{0.20}$	$\mathbf{\Gamma}_{0.30}$
Banknote	0.420	0.404 ↓	0.404 ↓	0.404 ↓	0.454 ↑	**0.506** ↑	0.504 ↑
Seeds	0.861	0.833 ↓	0.833 ↓	0.833 ↓	**0.886** ↑	0.857 ↓	0.811 ↓
Wine	0.875	**0.906** ↑	0.906 ↑	0.906 ↑	0.848 ↓	0.771 ↓	0.794 ↓
Votes	0.714	0.714	0.714	0.714	0.719 ↑	**0.730** ↑	0.722 ↑
Breast Cancer	**0.763**	0.755 ↓	0.755 ↓	0.755 ↓	0.679 ↓	0.506 ↓	0.382 ↓
Synthetic Control	**0.482**	0.482	0.482	0.482	0.000 ↓	0.000 ↓	0.000 ↓

Table 4.8: Benchmark datasets' Success Ratios using $w = 2$ and h_{xp}.

	MDH	$\text{MDH}_{\mathbf{\Gamma}_{MS}}$			$\text{MDH}_{\mathbf{\Gamma}_{H}}$		
		$\mathbf{\Gamma}_{0.10}$	$\mathbf{\Gamma}_{0.20}$	$\mathbf{\Gamma}_{0.30}$	$\mathbf{\Gamma}_{0.10}$	$\mathbf{\Gamma}_{0.20}$	$\mathbf{\Gamma}_{0.30}$
Banknote	0.362	0.395 ↑	0.385 ↑	0.385 ↑	0.386 ↑	0.405 ↑	**0.458** ↑
Seeds	0.892	**0.943** ↑	0.943 ↑	0.943 ↑	0.914 ↑	0.914 ↑	0.806
Wine	0.833	0.938 ↑	0.875 ↑	0.844 ↑	**0.938** ↑	0.875 ↑	0.844 ↑
Votes	0.718	0.731 ↑	0.706 ↓	0.695 ↓	**0.726** ↑	0.674 ↓	0.603 ↓
Breast Cancer	**0.763**	0.645 ↓	0.528 ↓	0.000 ↓	0.645 ↓	0.435 ↓	0.000 ↓
Synthetic Control	**0.482**	0.482	0.482	0.482	0.000 ↓	0.351 ↓	0.356 ↓

4.4.1 Standard MDH

Results from the standard MDH ($w = 0$) solution (Table 4.3) using the heuristic bandwidth were rather poor. MDH was able to learn meaningful clusters for the *Votes* and *Synthetic Control* datasets. While $\text{MDH}_{\mathbf{\Gamma}_{H(0.10)}}$ improved results for the *Votes* dataset, $\text{MDH}_{\mathbf{\Gamma}_{MS}}$ did not improve results. When the $\text{MDH}_{\mathbf{\Gamma}_{H}}$ outperforms $\text{MDH}_{\mathbf{\Gamma}_{MS}}$ it is essentially by chance, reassigning values in a way that is not intended by its formulation.

Applying the full bandwidth improved the MDH clustering performance relative to the heuristic bandwidth, with exception to the *Synthetic Control* results (Table 4.4). There were instances in which $\text{MDH}_{\mathbf{\Gamma}_{H}}$ improved the clustering solution; *Seeds*, *Wine* and *Votes* datasets but $\text{MDH}_{\mathbf{\Gamma}_{MS}}$ did not increase the quality of clustering for any of the six datasets.

The MDH solutions using the experimental bandwidth (h_{xp}) yielded the highest SR for the *Banknote*, *Seeds* and *Votes* datasets (Table 4.5). For every solution, both $\text{MDH}_{\mathbf{\Gamma}_{MS}}$ and $\text{MDH}_{\mathbf{\Gamma}_{H}}$ were able to improve the quality of clustering when $L = 0.10$. There were instances in which increasing L above 0.10 negatively impacted the quality of clustering for each enhancement method.

Consider the *Banknote* dataset. MDH failed to locate the majority of any cluster within this dataset when the heuristic and full bandwidth were applied. When the bandwidth was set using the experimental approach, the quality of clustering increased. $\text{MDH}_{\mathbf{\Gamma}_{MS}}$ and $\text{MDH}_{\mathbf{\Gamma}_{H}}$ were able to enhance this solution when $L = 0.10$. Figure 4.5 illustrates the densities extracted from the MDH solution for the *Banknote* dataset for each bandwidth, with $L = 0.30$ region indicated in green and the maximum feasible region represented as blue dashed lines. The heuristic and full bandwidths were larger than the experimental bandwidth. These larger bandwidths resulted in MDH solutions that clustered a relatively greater number of observations to one cluster compared to the experimental bandwidth MDH solution (Figure 4.5). Smaller bandwidth values resulted in higher quality, binary partition of the Banknote dataset.

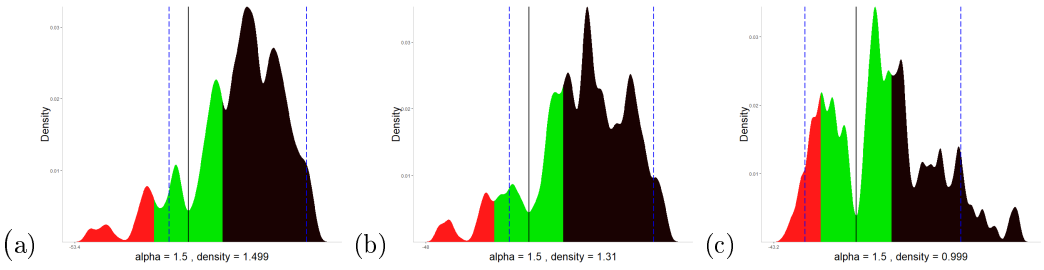


Figure 4.5: The *Banknote* data's MDH solution density plots; using the heuristic (a), full (b) and experimental (c) bandwidths. The red, green and black areas represent cluster 1, Gamma region and cluster 2 respectively.

4.4.2 Restricted MDH

Results from the restricted MDH ($w = 2$) solution (Table 4.6) using the heuristic bandwidth were relatively better than those obtained using the standard MDH approach. MDH was able to produce meaningful clusters for all but the *Wine* dataset. $\text{MDH}_{\mathbf{\Gamma}_H}$ and $\text{MDH}_{\mathbf{\Gamma}_{MS}}$ markedly improved the hyperplane solution for the *Banknote* dataset, even in instances where $L = 0.30$. However, there were occasions in which both methods degraded the hyperplane solution, more so as the value of L increased.

Results from improving the hyperplane solution using the full bandwidth were mixed (Table 4.7). MDH solutions were relatively similar to that obtained using the heuristic bandwidth. There were many instances in which $\text{MDH}_{\mathbf{\Gamma}_H}$ and $\text{MDH}_{\mathbf{\Gamma}_{MS}}$ deteriorated the original MDH cluster quality. $\text{MDH}_{\mathbf{\Gamma}_{MS}}$ did improve the *Wine* dataset's hyperplane solution, but $\text{MDH}_{\mathbf{\Gamma}_H}$ did not.

Applying the experimental bandwidth, the restricted MDH yielded cluster solutions similar to that obtained from applying the full bandwidth. In this setting, $\text{MDH}_{\mathbf{\Gamma}_H}$ and $\text{MDH}_{\mathbf{\Gamma}_{MS}}$ more often improved the hyperplane solutions (Table 4.8). This reiterates the importance of setting a proper bandwidth. The improvements to MDH were best when $L = 0.10$.

Standard MDH failed to locate the majority of any cluster using the heuristic and full bandwidth setting when clustering the *Banknote* dataset. Restricting the final solution plane to reside between the two largest modes increased performance when utilising these bandwidths (Figure 4.6(a and b)). The increase in clustering quality is most likely due to defining a hyperplane solution which yields a more balanced binary partition.

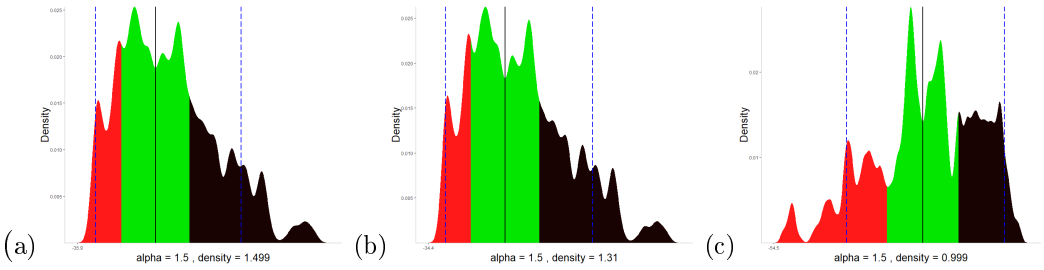


Figure 4.6: The *Banknote* data's restricted MDH ($w = 2$) solution density plots; using the heuristic (a), full (b) and experimental (c) bandwidths. The red, green and black areas represent cluster 1, Gamma region and cluster 2 respectively.

4.4.3 Benchmark test summary

Overall, restricting the hyperplane location to reside between the two largest estimated modes ($w = 2$) generally resulted in better clusters (Figure 4.7) relative to the standard MDH ($w = 0$) solutions. This is likely due to the restricted MDH solution producing relatively balanced clusterings compared to the standard version. Overall, the additional constraint that a separator must be located at the minimum density between prominent modes displayed promising results.

It is clear that the choice of bandwidth impacts the quality of clustering. Unfortunately, no single bandwidth selection rule was found to unequivocally outperform all others. However, the experimental bandwidth did present some promising results, especially when considering improving a hyperplane using MS (Figure 4.8). The choice of L was shown to effect the quality of clustering. On average, it appears that L values above 0.30, resulted in degrading the MDH solution when applying the single step gradient heuristic (Figure 4.9). Mean Shift had less variability when considering the Success Ratios across the various Gamma region sizes.

Overall, extending the linear hyperplane using MS can improve cluster quality. It was found that the single step reassignment approach could increase the quality of the hyperplane solution but results were highly variable. In cases where each enhancement degraded the solution, the heuristic did so at a greater extent than MS. From the benchmark tests results it appears that an ideal Gamma region is between $L = 0.10$ and $L = 0.20$.

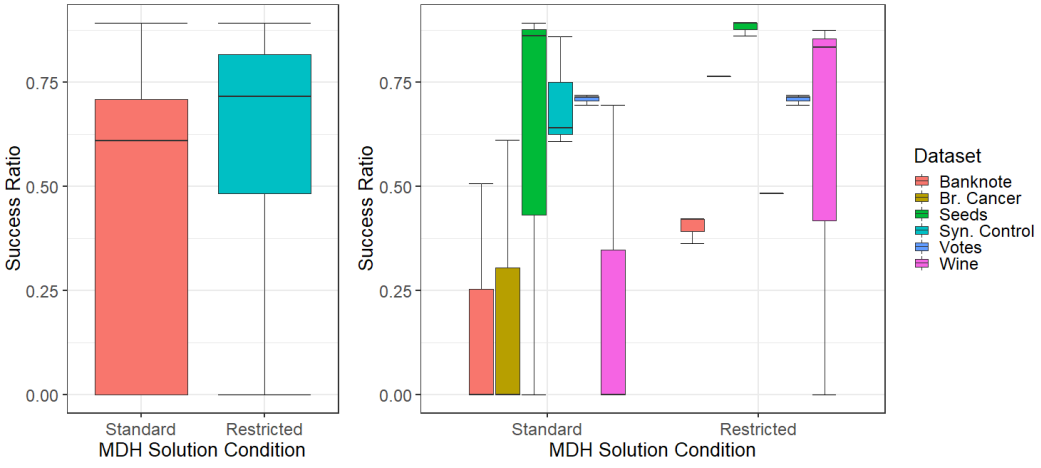


Figure 4.7: Boxplots of Success Ratios for standard ($w = 0$) and restricted ($w = 2$) MDH solutions overall (left) and per dataset (right).

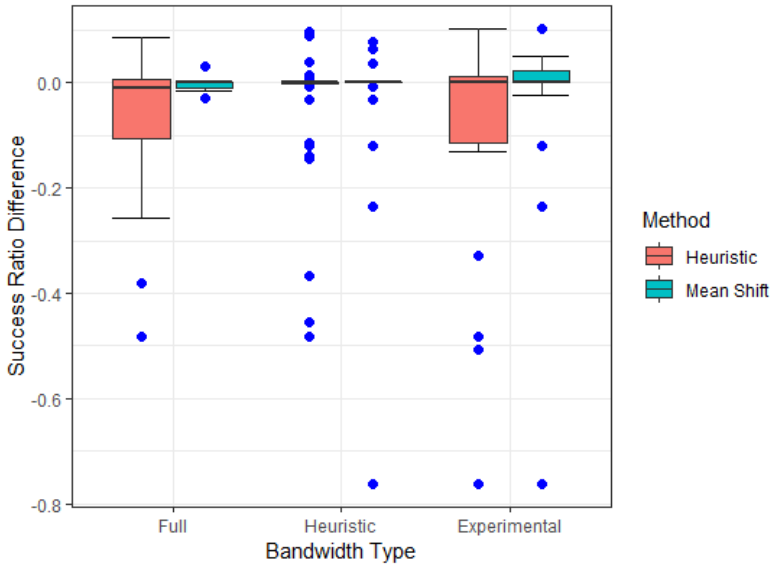


Figure 4.8: The change in MDH Success Ratio's across various kernel bandwidths per Mean Shift and the single step reassignment procedures. Zero indicates the SR of the original MDH solution.

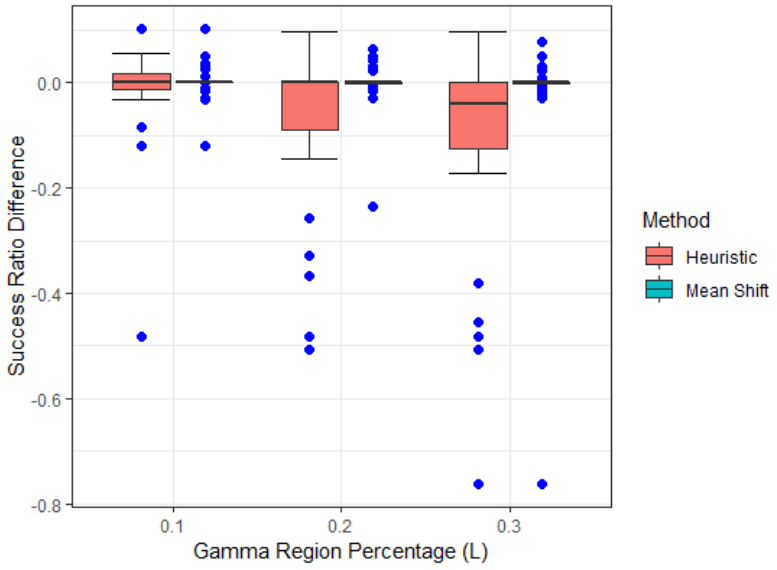


Figure 4.9: The change in MDH Success Ratio's across various Gamma region sizes per Mean Shift and the single step reassignment procedures. Zero indicates the SR of the original MDH solution.

4.5 Summary

The main limitation of MDH is that it defines clusters using a linear separator. To remove the limitation associated with a linear hyperplane, a novel approach was presented whereby a collection of objects around the hyperplane are re-assigned using a flexible clustering procedure. Mean Shift was presented as a capable technique to improve the MDH solution. A benefit from MS is that it can be applied to a single point, whereas a full density clustering requires the entire dataset. While MS is a sensible technique for reassigning objects, the computation involved to assign an object can be burdensome. This is exacerbated given larger datasets. A faster single step gradient approach was presented. The heuristic reassigns observations based on the initial gradient of a probability density function evaluated at each object.

Applying the MS enhancement to the *Type B* dataset MDH solution illustrated the process of superimposing MS cluster assignments onto the MDH solution using various values of L. We illustrated that the gradient ascent trajectories associated with each point generally did not change direction in relation to the hyperplane. This laid the foundation for the single step gradient heuristic approach. We showed that there is a relationship between the value of L and accuracy when applying the enhancements to the hyperplane solutions. Increasing L can lead to poor clustering solutions relative to lower values. Also, larger L values resulted in the heuristic reassigning points in a way that was not intended by its formulation. When considering a candidate region of points around a hyperplane solution we suggest that an ideal value for L is between 0.10 and 0.20.

We presented an additional constraint to the location of an MDH solution. It was motivated that a solution plane that separates prominent modes will improve cluster quality. With the added constraint, the feasible region around the mean can be increased and consequently the number of projection pursuit iterations. This increases the possibility of locating an optimal low-density separator. It was found that restricting the hyperplane to an interval between the two largest modes increased cluster quality relative to the standard MDH benchmark dataset solutions.

MDH, MDH$_{\mathbf{\Gamma}_{MS}}$ and MDH$_{\mathbf{\Gamma}_{H}}$ were applied to a set of benchmark datasets. It was found that the bandwidth and L parameters greatly impacted the final clustering solutions. We compared the heuristic, full and experimental bandwidth selection rules. Using the experimental bandwidth resulted in relatively better solutions in some instances. However, it cannot be said that this criterion is best overall.

We have shown that Mean Shift and its heuristic counterpart can improve MDH solutions but results depend greatly on the choice of bandwidth and Gamma region size. In the next chapter, improving the hyperplane clustering solution is evaluated as a tool for image processing.

Chapter 5

Application in Image Segmentation

5.1 Introduction

Image segmentation is a field within computer vision which seeks to automatically separate objects in a picture similarly to how the human visual system does. One basic task for image segmentation is to discriminate between the foreground and background of an image (Ballard and Brown, 1982). In order to understand how image segmentation works, it is helpful to understand the underlying data structure of images. Computers recognise pictures by associating each pixel with a level of intensity ranging from zero to one (255 for 8-bit encoded images). There are two main types of images, grayscale and colour. Grayscale pictures are represented by one channel (vector) which indicates the brightness level of each pixel. Colour images are most often represented by three intensity channels; red, green and blue (RGB). Points represented by zeros ([red=0, green=0, blue=0]) appear in a picture as black, ones ([1, 1, 1]) indicate white pixels, while all other combinations of RGB values represent the remaining colours. For example, pixels represented by [0.64, 0.16, 0.16] are brown and [1, 0.64, 0] are orange in colour. Clustering algorithms utilise the relational structure of pixel intensities to separate objects within an image.

As an initial step, each pixel location is indexed in order to maintain image structure (see Figure C.1). When images are of high resolution or contain many pixels, using prototypes is recommended in order to reduce computational expense. This is not always feasible as some algorithms require the entire dataset to produce a solution such as DBSCAN. Using a subset of the data is only possible for methods which produce a model from which predicted classes can be derived, such as MDH or K-means clustering. Besides the compulsory indexing, it may be advantageous to transform an image before applying a clustering algorithm. Decorrelation stretch (DCS) is one method which disperses pixel intensities so as to create an image which can be more

easily segmented (Alley, 1999). Minimum Density Hyperplane clustering can benefit from such a transformation.

MDH segments an image by associating distinct clusters based on estimated densities. For example, consider a picture consisting of a red square (focal object) atop a green background (Figure 5.1 (a)). Given the square is filled with pure red, its associated pixels densely populate the point [red=1, green=0 , blue=0]. The pixels representing green are concentrated at [0, 1, 0]. These dense regions can be separated using a plane defined by the equation, 1(red) - 1(green) + 0(blue) = 0 or $\boldsymbol{v}^\top \mathbf{X} = 0$, where $\boldsymbol{v}^\top$=[1, -1, 0]. This decision boundary equation assigns positive values to the red pixels and negative to the green pixels (Figure 5.1 b and c). Using this plane equation successfully segments foreground from background. Natural images will not be as clearly divided, instead pixels within the colour space will be more dispersed. However, objects whose colours do not vary greatly will be reasonably concentrated around some combination of RGB values. A probability density estimate from these pixels will capture this information in the form of a high density point at this location (a mode of the estimated density). Other dominant colours in the picture will produce similar modes around different RGB combinations. MDH attempts to find the equation of a plane which will separate these modes as well as possible, much like the plane used to separate the red square from its green background. More accurately, MDH seeks a hyperplane solution which minimises the integrated density along a continuous empirical probability density function(Pavlidis *et al.*, 2016).

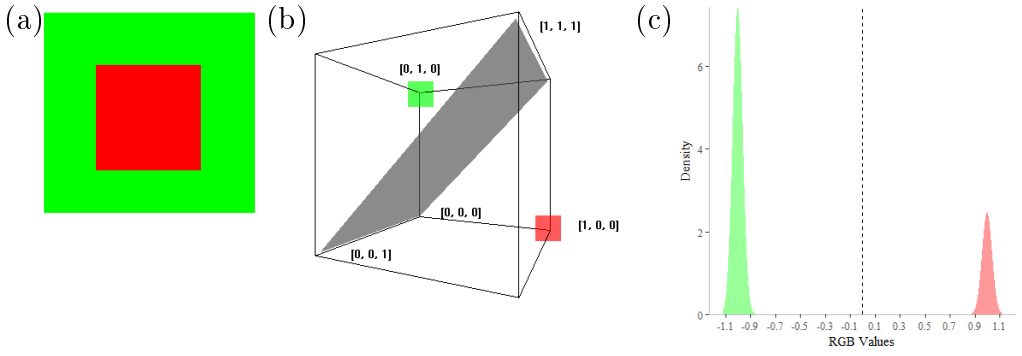


Figure 5.1: Image of a red square atop green background (a) with associated scatter plot of pixels in three dimensions (b) and density estimate of $\boldsymbol{v}^\top \mathbf{X}$ projection with decision boundary at zero indicated by the dashed line (c).

The remainder of this chapter is organised as follows: A simple technique which frequently allows for clearer separation of pixel intensities is presented as a viable pre-processing tool. This is followed by a practical application of image segmentation using MDH. We then manually segment an image using

principal components to underscore the importance of pre-processing an image. Then MDH is applied to a pre-processed image and compared to the non-transformed solution, whereafter an extension to the solution plane is applied to the pre-processed image using $\text{MDH}_{\mathbf{\Gamma}_{MS}}$ and then compared to its heuristic counterpart $\text{MDH}_{\mathbf{\Gamma}_{H}}$. This is followed by a comparative study between K-means, Max Margin Clustering (MMC), MDH, $\text{MDH}_{\mathbf{\Gamma}_{MS}}$ and $\text{MDH}_{\mathbf{\Gamma}_{H}}$ on a set of images from the Berkeley Segmentation Dataset (Martin *et al.*, 2001). This chapter concludes by summarising key results. Interested parties can further explore image segmentation using MDH and $\text{MDH}_{\mathbf{\Gamma}}$, interact with 3-dimensional scatter plots and generate manual separating planes online at *https://jacobbradleykenyon.shinyapps.io/Ch4_Enhanced_MDH*.

5.2 Pre-processing Images

There is a plethora of image processing techniques available; vector mode filters, dilation operators, erosion operators and contrast adjustments, only to name a few (Davies, 2012). When applying Minimum Density Hyperplane clustering, focus should be on methods which optimally disperse pixel intensities in such a way as to assist in defining a low-density separating plane. One such method is known as *decorrelation stretch*.

Decorrelation stretch enhances the colour differences amongst pixels. Principal component transformation provides a straight forward method to remove correlation within an image (Alley, 1999). To begin, the data matrix ($\mathbf{X}$) is transformed via normalised eigenvectors ($\mathbf{U}$) derived from the covariance ($\mathbf{S}$) eigendecomposition (Equation 5.2.2). With that, the decorrelated colour channels (Equation 5.2.2) can be stretched.

$$\mathbf{S} = \mathbf{U}\mathbf{D}\mathbf{U}^{\top} \tag{5.2.1}$$

$$\mathbf{X}^{*} = \mathbf{X}\mathbf{U} \tag{5.2.2}$$

Stretching RGB channels increases the contrast and object disparity within an image. Essentially, stretching amounts to rescaling pixel values to enforce an observed range which spans all possible values (Davies, 2012). Equation 5.2.3 illustrates the point-wise process of transforming decorrelated values (x^{*}_{ij}) within a colour channel dependent on a desired range for each vector ($\mathbf{x}^{*}_{j}$), viz. 255 for 8-bit RGB encoded images.

$$x^{DCS}_{ij} = \left[\left(x^{*}_{ij} - \min(\mathbf{x}^{*}_{j}) \right) \times \frac{\text{desired range}}{\left(\max(\mathbf{x}^{*}_{j}) - \min(\mathbf{x}^{*}_{j}) \right)} \right] \tag{5.2.3}$$

As a general rule, pre-processing data in a sensible manner can greatly improve final results. There is an abundance of transformation techniques available to analysts. Decorrelation stretch is applicable for image data but

can be extended to other structures, dependent on the domain within which the data are collected. The following section elaborates on the application of MDH and MDH$_{\mathbf{\Gamma}}$ in image segmentation while illustrating the usefulness of decorrelation stretch.

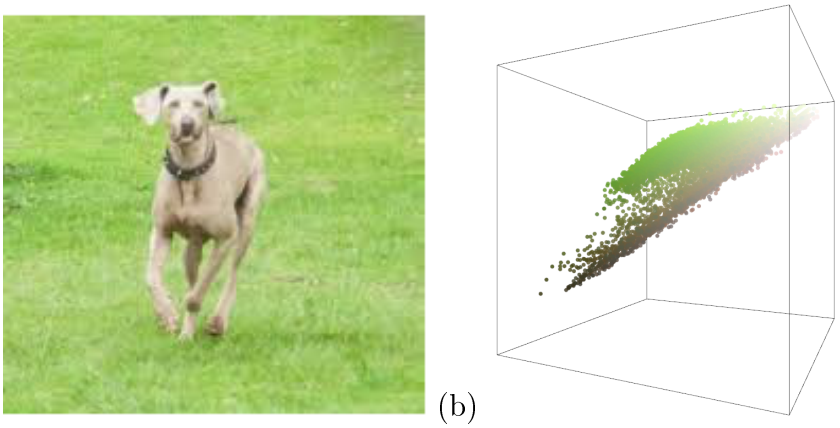
5.3 Image Segmentation Using MDH

Successfully segmenting an image using MDH requires that objects within the foreground consist of colour combinations that densely populate a region different to that of the background. This subsection illustrates; how minimum density decision boundaries identify clusters within an image, results from manually separating an image, MDH results from pre-processing an image and the application of gradient ascent to enhance the MDH solution, using an image of a weimaraner (`https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcT2w5H7Jpr0_DJkpLpCZE80T7aiRzeJtyjhH6wQpPwwUfg8XRbW-w`).

Figure 5.2 (a) is an image of the dog with its associated scatter plot of pixel intensities in RGB space (Figure 5.2 (b)). Each point within the scatter plot is colour matched to the image for clarity. Two tones are dominant in the image, namely green and brown. Notice that the direction with the largest amount of variation explained is defined by the various tones of each dominant colour. The weimaraner image comprises of darker and lighter shades of brown. Similarly, the field has varying tones of green. While further exploring the scatter plot, one can imagine an optimal plane which separates brown from green toned pixels which is parallel to the direction of the first principal component. This is equivalent to segmenting the dog (foreground) from the field (background), since most of the dog is brown while the field is primarily green. Ideally, MDH will locate this separator to segment the image.

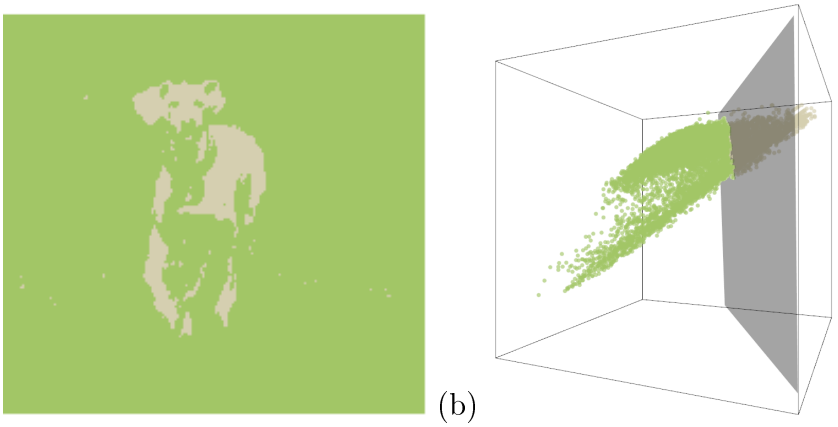
Initial MDH results segmented lighter from darker tones within the picture (Figure 5.3(a)). While most of the lighter tones are associated with the dog, they are also present within the field. This creates a sub-optimal cluster solution. It is apparent that the decision boundary failed to fully segment brown from green (Figure 5.3(b)). This has led to a lack of full focal definition, since dark brown tones associated with dog shadows were assigned to the background (Figure C.2).

Recall that MDH utilises projection pursuit initialised on the first principal component, whereby iterative density estimates of projections are used to seek an optimal low-density linear hyperplane solution. A cause for the sub-optimal separating plane is due to the relatively larger amount of variation in the first principal component compared to the second, which inhibits optimal rotation. This is evident when comparing the final projected density against the density estimate of the data projected onto the second principal component.



(a) (b)

Figure 5.2: Image of the dog (a) with associated scatter plot of pixels in three dimensions (b), colour matched for clarity.



(a) (b)

Figure 5.3: Minimum Density Hyperplane solution (a) and associated scatter plot of pixels (b), colours represent the average RGB channel intensities from clusters assigned by separating plane.

Investigating the density associated with the final MDH univariate projection indicates that the hyperplane did in fact fail to converge at a low-density separating plane (Figure 5.4(a)). Note that as the feasible region expands over the mean, a minimum density is sought such that it lies between prominent modes, where the first and second largest estimated mode is associated with a point on $\boldsymbol{v}$ at approximately -1.5 and -2.2. While there are points on $\boldsymbol{v}$ with lower densities between these two modes, there are relatively smaller densities opposite the largest mode. Since these relatively smaller densities are not within the interval dictated by the two largest modes [-2.2, -1.5], the solution reverts back to the last known acceptable solution, which is indicated as a red line in Figure 5.4(a) with the maximum feasible region indicated as

dotted black lines. Ultimately, projection pursuit was unable to identify the previously postulated optimal separator. Comparing modes between the density functions related to the solution (Figure 5.4(a)) and that from the second principal component projection (Figure 5.4(b)) illustrates why projection pursuit failed. The variation along the first principal component is much larger than the second. This leads to the first principal component projection density having a peak mode far lower (≈ 2.5) than the second principal component (> 10). This inhibits optimal rotation since the density associated with maximum variability orthogonal to the projection vector is far lower than that contained within the second principal component. The following section illustrates a manual image segmentation solution by projecting onto the second principal component.

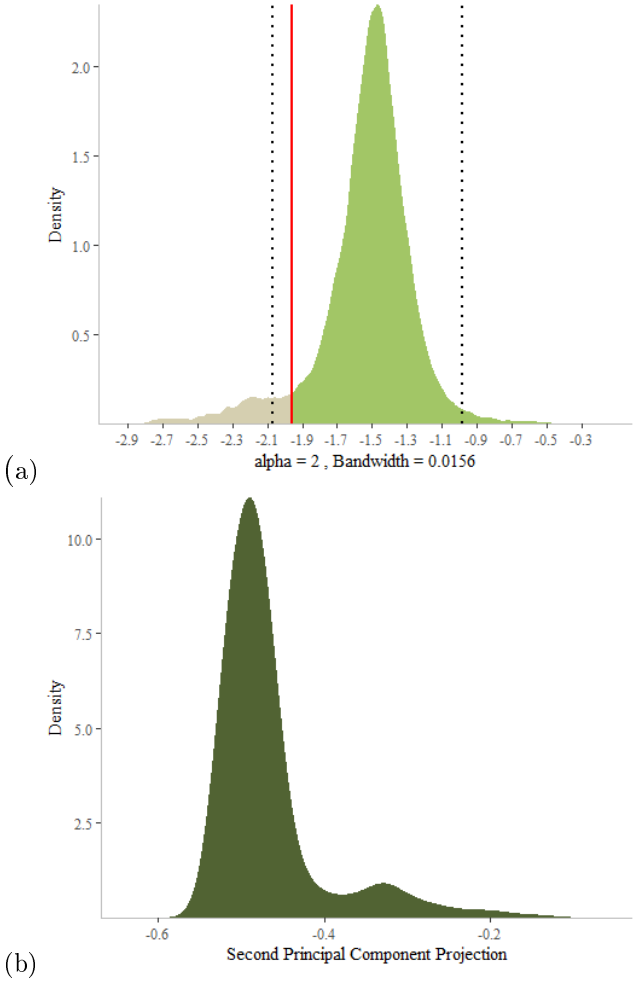


Figure 5.4: Density of the final univariate projection of the MDH solution (a) and density resulting from projecting onto the second principal component (b). The hyperplane is the red line and the maximum feasible region is indicated as dashed lines.

5.3.1 Manual solution

It is evident thus far that an ideal separating plane segments green tones from brown. This solution plane lies orthogonal to the second principal component. One could manually segment the image by estimating the density of the image data projected onto the second principal component axis and assign clusters based on the pixel locations relative to the minimum density between two dominate modes. Evaluating the density estimate, it appears an ideal separating plane is located at -0.38 (Figure 5.5(a)). The manually segmented image (Figure 5.5(b)) vastly improves upon the initial MDH algorithm solution. However, not all images will have a clear solution from projecting once onto a principal component axis and it is time consuming to manually search along principal axes for optimal densities. MDH solves this problem in an automatic way. If the variances between principal components are not drastically different, then projection pursuit will be able to rotate the data and find a projection index from which an optimal low-density separator is defined. Pre-processing an image with decorrelation stretch before applying MDH may improve results since stretching the image along all principal axes reduces the disparity between principal component variances.

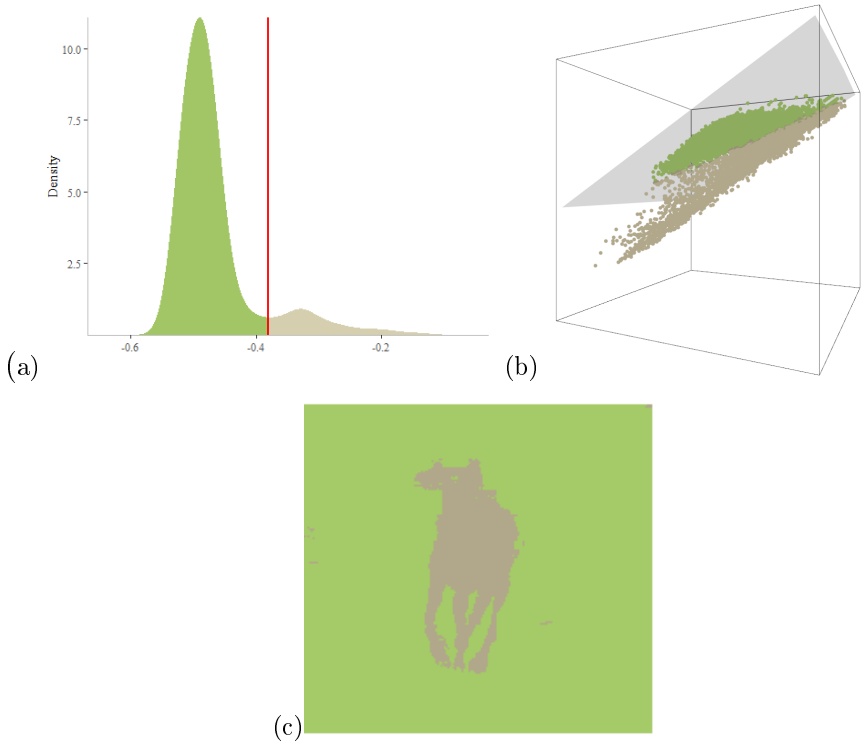
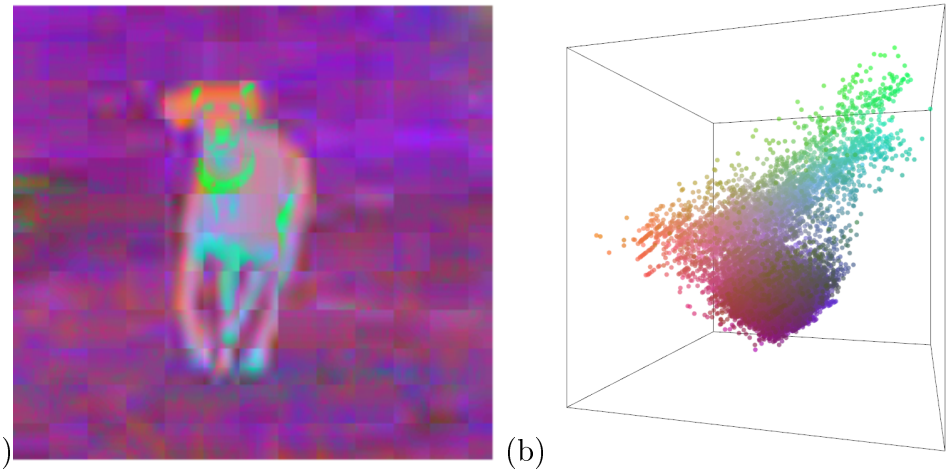


Figure 5.5: Solution (c) using density estimate of projection onto the second principal component axis (a) with the manually chosen plane in red and with clusters in RGB space(b).

5.3.2 Decorrelation stretch and MDH

Decorrelating and stretching an image disperses distances between pixels within the colour space by maximising the variance along each principal component. Figure 5.6(a) illustrates the transformed image of the dog. Observed pixel intensities are relatively further apart (Figure 5.6(b)) compared to the non-transformed image. Within the transformed space there are two apparent cluster regions, one consisting of darker colours and another with lighter toned pixels. Exploring the scatter plot, one can imagine a hyperplane which separates the darker tones in the lower half from lighter ones within the upper half as being an optimal solution. Given the dispersion of pixels, MDH should more easily locate this plane when clustering the image.

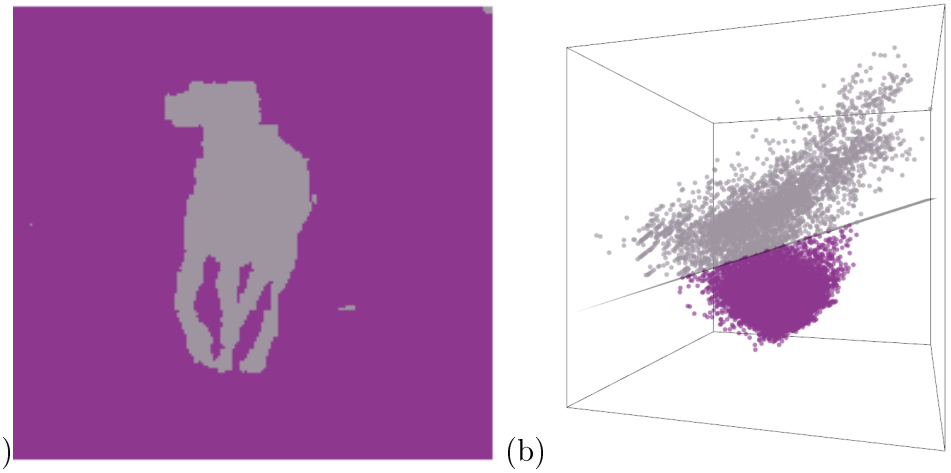


(a) (b)

Figure 5.6: Decorrelated and stretched dog image (a) with associated scatter plot of pixels (b), colour matched for clarity.

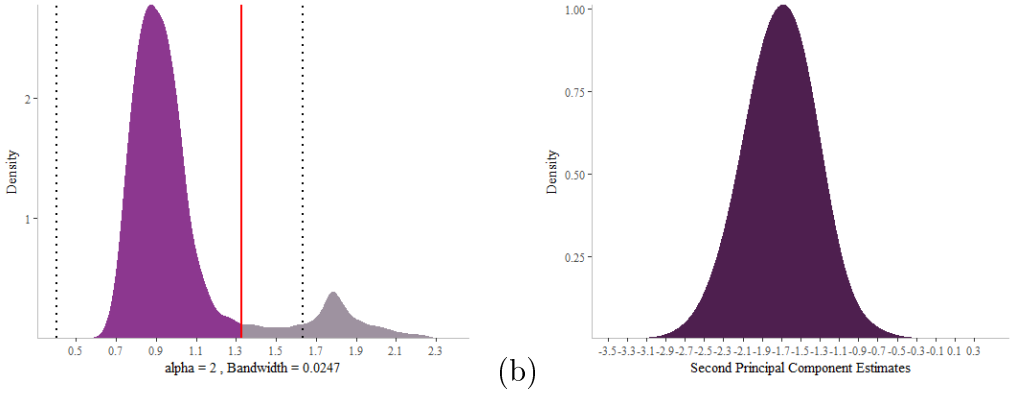
Pre-processing has led to a more desirable MDH solution (Figure 5.7(a)), successfully identifying the focal object within the image. These results are echoed in the scatter plot (Figure 5.7(b)), producing a solution plane located within the region previously postulated as optimal. Decorrelating and stretching the image has resulted in a final kernel density estimate which indicates a second modal point (Figure 5.8(a)). While the solution plane has improved results, it did not converge at a local minimum. This is because points that lay equidistant to the mean are associated with lower density estimates and are not between two modes. Thus, the final solution reverts to the lowest density between two modes. Also, notice the mode of the density estimate associated with the second principal component projection is less than the first principal component projection (Figure 5.8(b)). Herein lies the benefit of DCS, stretching the image along all principal axes allows projection pursuit to optimally rotate towards an ideal density to segment along. This is made possible by

reducing the disparity across principal components and as such allows projection pursuit based on minimum density to more easily rotate the data. Next we apply gradient ascent to a collection of pixels in a neighbourhood around the solution hyperplane to further improve the DCS-MDH results.



(a) (b)

Figure 5.7: Minimum Density Hyperplane solution from decorrelated and stretch dog image (a) with its associated scatter plot of pixels (b).



(a) (b)

Figure 5.8: Density of the final univariate projection of the MDH solution (a) and the second principal component projection density (b) from the decorrelated and stretched dog image. The hyperplane is indicated in red, with the feasible region contained within the dashed lines.

5.3.3 Improving the linear hyperplane

Building upon the DCS-MDH solution, a candidate region of pixels around the hyperplane is identified for possible cluster reassignment with $L = 0.15$ (Figure 5.9 (a)). With a defined reassignment region, gradient ascent can be applied as a method to improve the hyperplane solution.

Using Mean Shift, each object within the Gamma region is assigned to a cluster based on its location within a modal attraction basin. A kernel based estimate of the density is constructed using the same bandwidth as that used to obtain the MDH solution. The domain of attraction of a mode is a feature of this density function, attraction basin. Each attraction basin is represented by a mode (modal point). If the modal point of a pixel resides in a new cluster it is reassigned. Evaluating Figure 5.9, it appears that the modal points for many of the grey pixels within the Gamma region lie in attraction basins which have estimated modes that are below the decision boundary, resulting in reassigning many grey clustered pixels to purple (Figure 5.9(c)). While the solution is ideal, a large amount of computing time was required since pixels are assigned only after converging to their modal point. One way to reduce computation is to assign each pixel to a cluster based on the initial gradient ascent trajectories.

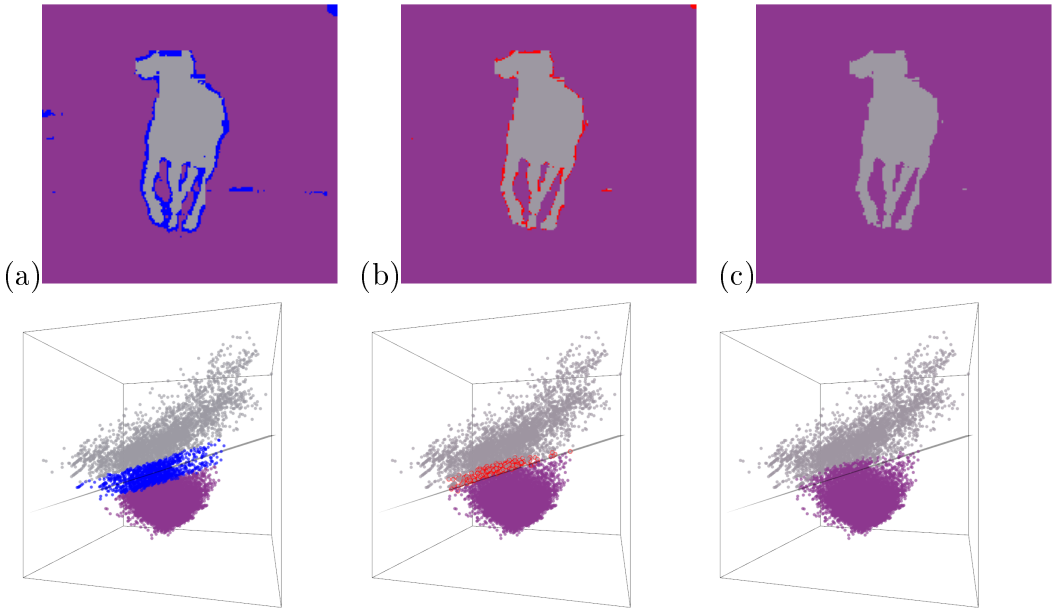


Figure 5.9: Mean shift adjusted MDH dog image with associated scatter plot of points around solution plane: $\mathbf{\Gamma}_{0.15}$ reassignment region indicated by blue coloured points (a), reassigned points indicated in red (b) and final adjusted solution (c).

The single step first order kernel derivative approach (utilising the same bandwidth as before) greatly reduces computation compared to the Mean Shift approach. Recall that the heuristic procedure calculates the inner product of $\boldsymbol{v}$ and $\nabla\hat{p}(\boldsymbol{\gamma})$ and assigns those pixels associated with positive values to the cluster above the hyperplane. Using the heuristic approach resulted in a few extra purple pixels being reassigned to the grey cluster (Figure 5.10) compared to MS. Overall, the heuristic approach is an acceptable alternative as it produces similar results in a fraction of the time while inducing a small amount of image noise.

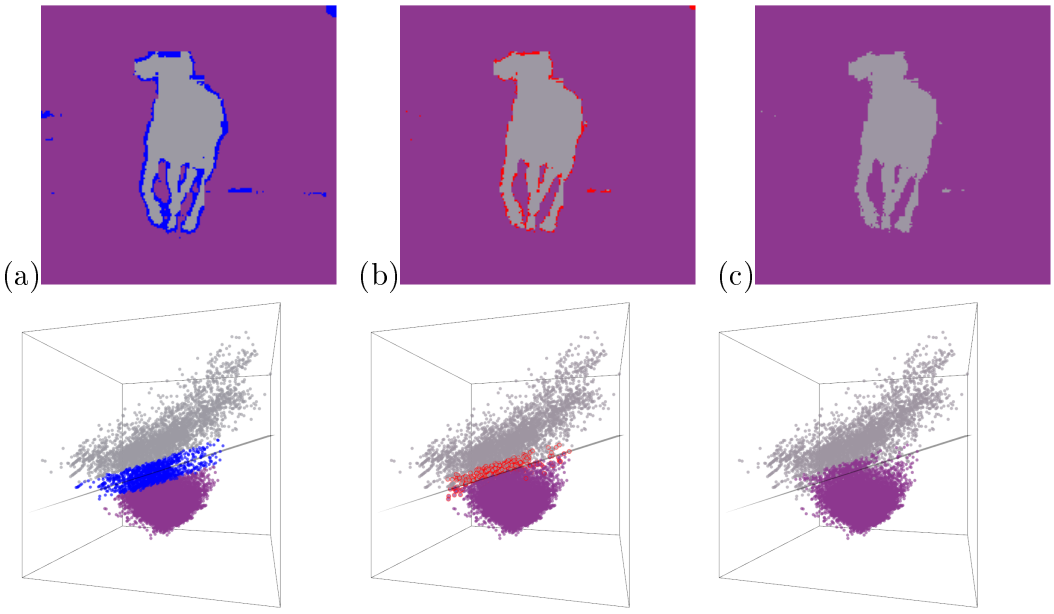


Figure 5.10: Heuristic adjusted MDH dog image with associated scatter plot of points around solution plane: $\boldsymbol{\Gamma}_{0.15}$ reassignment region indicated by blue coloured points(a), reassigned points indicated in red(b) and final adjusted solution(c).

5.3.4 Combining MDH enhancements

Thus far, MDH was applied directly to the non-transformed dog image with little success, an ideal solution plane was produced manually and a transformation was presented as an automatic technique to find the optimal solution. The DCS-MDH solution was then extended to a non-linear form using Mean Shift and its single step gradient heuristic counterpart. This subsection illustrates how to combine DCS and the enhancements to the solution plane while retaining original RGB colours.

To obtain an ideal solution plane, the image data is transformed using DCS. Within the DCS-RGB space, MDH locates the optimal projection and

hyperplane solution equation. From the solution, a region of points around the minimum density plane are then considered for reassignment using the Mean Shift heuristic approach. Figure 5.11 highlights the $L = 0.20$ Gamma region (a), values which were reassigned (b) and illustrates the final results (c) and the DCS-MDH heuristic solution converted to the original input space. Notice that Gamma region pixels are located around the edge of objects within the picture, reiterating the initial DCS-MDH solution was ideal. If the Gamma region consisted of observations not associated with edges, reassigning clusters may have had no impact or even deteriorated the solution. As a qualitative check, Gamma region points that do not lay along boundaries in the picture indicates a poor initial solution. Overall, one can utilise the DCS data and then simply map the final cluster labels to the original space. Results from the DCS-MDH heuristic applied to a Gamma region with $L = 0.20$ slightly improved upon the previous results when $L = 0.15$ (Figure 5.10). Increasing L to a 0.20 has led to a small amount of background noise reduction.

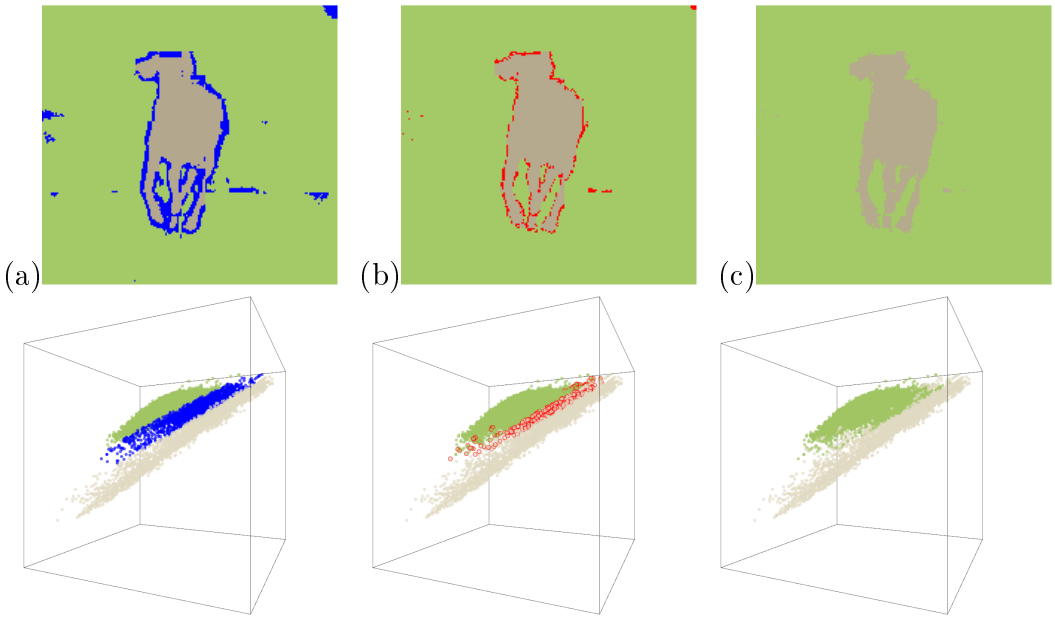


Figure 5.11: Decorrelated and stretched MDH solution mapped to original image colour space; $\mathbf{\Gamma}_{0.20}$ region indicated by blue coloured points (a), reassigned values in red (b) and final solution (c) with associated scatter plots below each image

5.3.5 MDH image segmentation summary

Initial MDH results of the non-processed picture produced a clustered image with relatively low focal definition. The variance explained along the first principal component was far greater than the variance explained within the second principal component. The first principal component of an image may generally have greater variance than the second component due to the nature of how colours shift tone from shadows to light. This was identified as the cause for the initial sub-optimal solution. DCS transformation was then presented as a tool to combat the inherent relative differences between the first and second principal components in RGB images. Dispersing pixels within RGB space before applying MDH, produced a quality separating plane. The DCS-MDH solution plane was then extended to a non-linear form by applying MS and its heuristic counterpart to a Gamma region with $L = 0.15$. The single step heuristic approach proved to be a viable alternative to MS, producing similar results to MS in a fraction of the time. Utilising a Gamma region with $L = 0.20$ and mapping the DCS solution back to the original image illustrated the overall process of segmenting an image. In the following section; a comparative study between MDH, MDH$_{\mathbf{\Gamma}}$, K-means and Maximum Margin Clustering (MMC) is undertaken. Therein each method is compared for their ability to discriminate the foreground from the background of an image. This study was undertaken using several pictures from the Berkeley image database (Martin *et al.*, 2001).

5.4 Comparison of Clustering Algorithms

A key goal of image segmentation is to identify main objects within a picture by separating foreground from background. Most cluster algorithms utilise dissimilarity matrices, whereby computation increases quadratically with the number of pixels and is often infeasible in R (Azzalini and Menardi, 2014). General density based methods are also computationally expensive and implementations often fail or require enormous amounts of time to segment a single image. Since DBSCAN and MS require full datasets in order to segment an image, these methods will not be considered. The image segmentation study will compare K-means, MMC, MDH and MDH$_{\mathbf{\Gamma}}$.

MMC is the unsupervised version of Support Vector Machine (Cortes and Vapnik, 1995, SVM) classification. MMC seeks a hyperplane which maximises the margin between clusters such that if SVM were subsequently applied, it would obtain the same boundary (Xu and Schuurmans, 2005). Since not all data are separable, MMC introduces a slack parameter that allows for a soft margin in the style of SVM. This technique has been found to outperform most density-based clustering methods (Xu *et al.*, 2005). This method is an ideal choice to compare with MDH because it also incorporates the notion of a low-density separator. Pavlidis *et al.* (2015) showed that the MDH solu-

tion converges to the maximum margin hyperplane solution as the bandwidth reduces to zero.

As an intial step, all images' pixel intensities were indexed. The datasets were then compressed using mean prototypes. This was achieve by considering a 25×25 Cartesian graph over each image and calculating the mean RGB values within each grid. This effectively reduced the number of pixels to 625 and greatly reduced the time complexity required by each clustering procedure. Furthermore, decorrelation stretch was applied to the mean prototype data. Two-way clustering was then applied to the DCS-transformed mean prototypes.

All two-way cluster analyses were applied using R. K-means was calculated using `kmeans` from the `stats` R-package (R Core Team, 2018). MMC was applied using `mmc` within `bmrm` R-package with $\lambda = 0.0001$, a parameter which controls the amount of regularisation during the SVM optimisation process (Prados, 2018). The feasible region for MDH was set with $\alpha = 1.5$, while the solution was restricted to an interval defined by the 5 largest estimated modes ($w = 5$). For MDH, different bandwidth selection rules were utilised for each image. The densities were estimated using either the heuristic, full or experimental bandwidths. The bandwidth used to obtain the MDH solution was also used to reassign pixels. MS and its heuristic counterpart were then applied to a Gamma region with $L = 0.20$ over the entire dataset. Each procedure's hyperplane solution equation were then used to map the prototype solution back the original image space. Figure 5.12 illustrates the image segmentation results applied to a selection of images from the Berkeley image database (Martin *et al.*, 2001). The first column contains the original picture and the second column contains a set of human segmented images. Interested readers can evaluate how each method clustered the dog image within Appendix C.4.

Reviewing results from Table 5.12, MDH arguably performed the *best*. MDH image segmentation results closely resembled the human segmented images of the elephants, fighter plane and the surfer. There were mixed results when enhancing the hyperplane solution. $\text{MDH}_{\boldsymbol{\Gamma}_{MS}}$ was superior to $\text{MDH}_{\boldsymbol{\Gamma}_{H}}$, given the heuristic procedure resulted in images that contained relatively more noise (viz., eagle, airplane, fighter plane and elephant pictures). Reassigning pixels once they had converged to their estimated mode resulted in images with refined focal definition with relatively less background noise compared to the MDH solution. To further compare each technique, we evaluated some statistics from a sample of the image segmentation results (Table 5.1).

The time required for an algorithm to process an image is a direct indication of usability (Table 5.1). K-means executed image segmentation fastest while $\text{MDH}_{\boldsymbol{\Gamma}_{MS}}$ was the slowest, requiring at least 3.5 minutes to segment an image. On average, MMC took approximately 30 times longer than MDH to segment the images in Table 5.1. Using MS to reassign Gamma region observations took nearly 50 times longer than the heuristic single step.

Figure 5.12: Binary image segmentation results from 2-means, MMC, MDH, MDH$_{\mathbf{\Gamma}_{MS}}$ and MDH$_{\mathbf{\Gamma}_{H}}$ procedures.

Table 5.1: Subset of image segmentation results from comparative study

	Solution	Time	Silhouette Coefficient
2-means		**0.015 secs.**	0.471
		**0.021 secs.**	**0.496**
		**0.199 secs.**	0.577
		**0.020 secs.**	0.543
MMC		1.585 mins.	**0.474**
		1.416 mins.	0.430
		1.228 mins.	**0.584**
		1.181 mins.	0.368
MDH		1.944 secs.	0.474
		7.558 secs.	0.415
		2.446 secs.	0.560
		2.708 secs.	**0.553**
$\text{MDH}_{\mathbf{\Gamma}_{MS}}$		5.849 mins.*	0.474 ↓
		3.944 mins.*	0.391 ↓
		17.565 mins.*	0.575 ↑
		3.523 mins.*	0.533 ↓
$\text{MDH}_{\mathbf{\Gamma}_H}$		6.886 secs.*	0.472 ↓
		6.476 secs.*	0.414 ↓
		16.875 secs.*	0.570 ↑
		5.249 secs.*	0.522 ↓

A sample of time taken to execute algorithm and silhouette coefficients. Each algorithm was applied to compressed images represented by a 25×25 grid of mean prototypes. *Best* results are indicated in bold.

* $\text{MDH}_{\mathbf{\Gamma}}$ represents time taken to evaluate $\mathbf{\Gamma}_{0.20}$ region over the full dataset and do not account for time to obtain original MDH solution.

The silhouette coefficient was computed using the mean prototypes and their associated cluster assignment. Recall from Chapter 2, silhouette coefficients values near one signify well clustered results. According to the silhouette coefficients, MDH clustered the surfer image best, 2-means clustered the elephant image best, while MMC had the highest quality of clustering for the fighter plane and scenic mountain pictures. $\text{MDH}_{\mathbf{\Gamma}_{MS}}$ and $\text{MDH}_{\mathbf{\Gamma}_{H}}$ improved the silhouette coefficient for the fighter plane image while decreasing for the other images. One can easily argue that the silhouette coefficient is misleading. Especially in the case which $\text{MDH}_{\mathbf{\Gamma}_{H}}$ improved the MDH silhouette coefficient for the fighter plane image where it clearly added a great amount of noise to the solution. While the silhouette coefficient may be able to provide some insight into the quality of clustering, the significance of this metric should be considered with care.

5.5 Summary

A key task in image segmentation is to identify the foreground of an image. DCS was introduced as a pre-processing technique which can assist in image segmentation by dispersing pixel intensities within RGB colour space. Comparing MDH results between the original and DCS image reiterated the benefit of such a pre-processing method for image segmentation. DCS transformations can improve MDH results by reducing the disparity between principal component variances, allowing projection pursuit to more easily rotate the data which increases the possibility of identifying an optimal projection vector.

$\text{MDH}_{\mathbf{\Gamma}}$ was then presented as the culmination of MDH re-evaluated within a Gamma region. Building on results from the DCS-MDH dog solution, a collection of pixels within a neighbourhood around the hyperplane were reassigned using gradient-based procedures. $\text{MDH}_{\mathbf{\Gamma}_{MS}}$ was found to improve image segmentation results. The heuristic approach substantially reduced reassignment computation and segmented the image similar to $\text{MDH}_{\mathbf{\Gamma}_{MS}}$.

The image segmentation comparative study indicated that MDH on average outperformed 2-means and MMC procedures. While $\text{MDH}_{\mathbf{\Gamma}}$ generally improved upon MDH, there were instances in which $\text{MDH}_{\mathbf{\Gamma}_{H}}$ introduced relatively large amounts of noise. $\text{MDH}_{\mathbf{\Gamma}_{MS}}$ improved the MDH solutions but required substantially greater amounts of time to reassign values than its heuristic counterpart. It was found that the silhouette coefficient did not yield consistent results when comparing how well each method segmented an image. Overall, MDH is a viable image segmentation tool. The choice of technique to improve the hyperplane solution is dependent on the task at hand and whether noise induced by applying $\text{MDH}_{\mathbf{\Gamma}_{H}}$ is considered acceptable.

Chapter 6

Conclusion

6.1 Summary

The aim of this study was to improve hyperplane solutions in the context of clustering. We introduced a novel approach that reassigned a collection of points within a neighbourhood of the hyperplane (Gamma region) using more flexible gradient-based methods. We presented the gradient ascent procedure, Mean Shift, as a sensible technique since it can be applied to one observation at a time and assigns points in a flexible, non-linear way.

Mean Shift reassigns each point within a Gamma region according to its associated attraction basin estimated mode in relation to the hyperplane. It was illustrated that the gradient ascent trajectory's associated with each object generally did not change direction with regards to the hyperplane solution. We utilised the relationship of the initial gradient direction to the hyperplane to formulate the single step gradient heuristic.

The single step gradient heuristic reassigns Gamma region observations based on the estimated slope of a function evaluated at each point in relation to the hyperplane solution. If the initial gradient points towards the hyperplane, then it is likely that the gradient ascent terminates opposite the low-density separator and the associated observation is thus reassigned to a different cluster.

An additional restriction to the location of a hyperplane solution on $\boldsymbol{v}$ was presented. This post-optimisation process restricts the hyperplane to an interval between w prominent modes. We motivated that this additional constraint can guard against the possibility of locating a hyperplane near the boundary of a density while also allowing for a larger feasible region from which to locate an optimal low-density separator via projection pursuit within MDH. While this showed promising results, it was highly dependent on the choice of bandwidth. Ultimately, choosing an inferior smoother will result in poor clustering solutions.

An empirical study was undertaken using a variety of UCI benchmark datasets to evaluate the clustering performance of MDH and its enhancements. It was found that restricting the MDH solution to reside between the two largest estimated modes, generally improved results. We also considered three different bandwidth selection rules; the heuristic, full and experimental bandwidths. The results indicated that no single bandwidth selection rule unequivocally outperformed any ot the others. When considering the Gamma region, it was found that sometimes larger values of L, deteriorated the performance of each enhancement method, more so for the single step gradient heuristic approach. We concluded that an ideal value of L is between 0.10 and 0.20.

An image segmentation study was undertaken to evaluate the performances of K-means, MMC, MDH, $\text{MDH}_{\mathbf{\Gamma}_{MS}}$ and $\text{MDH}_{\mathbf{\Gamma}_{H}}$ using a selection of images from the Berkeley image database. On average, we found that MDH was able to segment the images better than K-means or MMC. Results from enhancing the MDH solution were promising when considering the MS reassignment approach. $\text{MDH}_{\mathbf{\Gamma}_{MS}}$ solutions were found to be closer, on average, to the human segmented images. However this approach required the most time to arrive at a final clustering solution. $\text{MDH}_{\mathbf{\Gamma}_{H}}$ was notably faster but resulted in some segmented image solutions with considerably more noise than the MDH or $\text{MDH}_{\mathbf{\Gamma}_{MS}}$ solutions. Overall, MDH and the enhancements thereof are capable image segmentation tools that can discriminate between the foreground and background of an image.

6.2 Future Research

While the Gamma region was mainly utilised to enhance MDH solutions, it can also be used to improve other clustering and classification techniques which model decision boundaries. Furthermore, evaluating the region around a hyperplane may prove useful in determining an ideal bandwidth given the relationship between stability and low-density separators established by Ben-David and Von Luxburg (2008). This formed the basis of the experimental bandwidth selection rule. Conceptually, a bandwidth that produces a minimum density hyperplane with few points within the Gamma region is more ideal than that which contains many points. The experimental bandwidth showed promising results. The relationship between stability and low-density separators can further be exploited as a method to find an optimal value of k. We propose that a stable MDDC solution, given a value of k, is one which exhibits the lowest penalised average proportion of observations within the neighbourhoods of all low-density separators.

6.3 Conclusion

The proposed enhancement to the hyperplane solution showed promising results. The amount by which a hyperplane solution was improved using Mean Shift and its heuristic, greatly depended on specifying a proper smoothing parameter and reassignment region size around the low-density separator. Overall, we showed that by reassigning a collection of observations around a low-density separator using Mean Shift and a single step gradient heuristic can improve the final clustering solution.

Appendices

Appendix A

Cluster Analysis

A.1 Euclidean Distance Example

Euclidean distance is a common dissimilarity measurement used to perform cluster analysis. We can calculate Euclidean distances in order to cluster the simple example of persons weight and height that was introduced in Chapter 2. Consider persons a and b, where person $a = (weight_a, height_a)$ and person $b = (weight_b, height_b)$. The Euclidean distance between these objects is defined as:

$$d(a,b) = \sqrt{(weight_b - weight_a)^2 + (height_b - height_a)^2}, \tag{A.1.1}$$

$$= \sqrt{(120-65)^2 + (180-40)^2}, \tag{A.1.2}$$

$$= 68.01 \tag{A.1.3}$$

The computation of all other pairwise distances follow similarly as described above (Table A.1). Person a would be clustered with person e given the relatively small pairwise distance of 12.37 compared to persons b, c and d. Similarly, subjects b, c and d (highlighted in blue) would form a cluster since they have relatively low pair-wise distances compared to a or e.

Table A.1: Euclidean distance dissimilarity matrix.

Subject	**a**	**b**	**c**	**d**	**e**
a	0				
b	68.01	0			
c	57.01	11.18	0		
d	71.06	18.03	20.00	0	
e	12.37	59.06	47.89	60.11	0

Appendix B

Enhancing the hyperplane solution

B.1 Mean Shift Assignment

Figure B.1 represents the estimated modes from clustering *Type A* and *Type C* data types. An advantage of MS is that the user does not need to pre-define the number of clusters. However, the choice of bandwidth to use within the kernel density estimation is required and the choice of which is not trivial. The heuristic bandwidth selection rule was used to cluster the data. In order to compare MS with MDH, the minimum cluster size was set to 245 because it was found to produce a binary solution. Due to this constraint, elements associated with the upper-right estimated mode within Figure B.1(a) were joined to the nearest estimated mode according to centroid linkage hierarchical clustering using Euclidean distances. Similarly, *Type C* assigned elements from the outer ring to the centre sphere of data points.

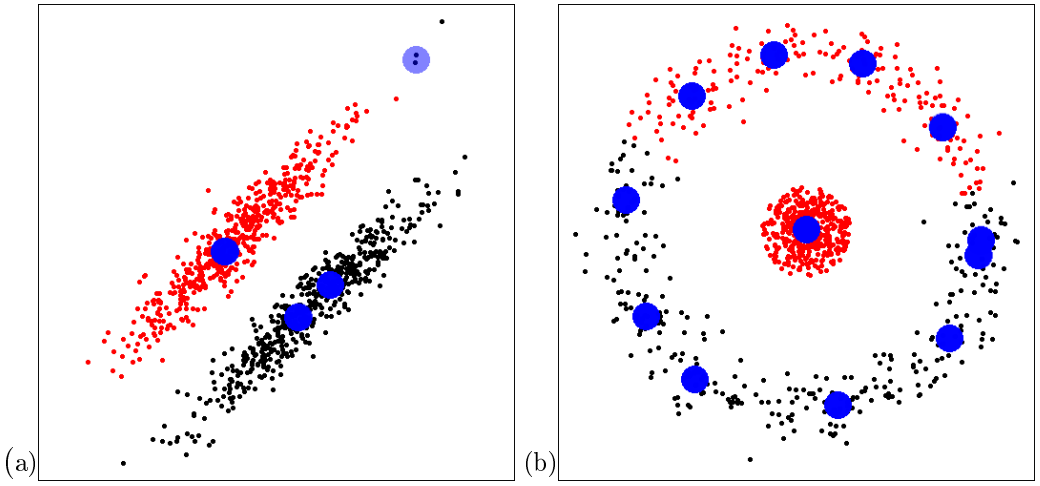


Figure B.1: Mean Shift assignment for *Type A* and *Type C* data types with local modes indicated as blue dots.

Figure B.2 illustrates MS clustering without the constraint of minimum cluster size. MS correctly discriminate the centre sphere of the *Type C* data from all other elements (a). The overlapping cluster type can not be segmented by MS.

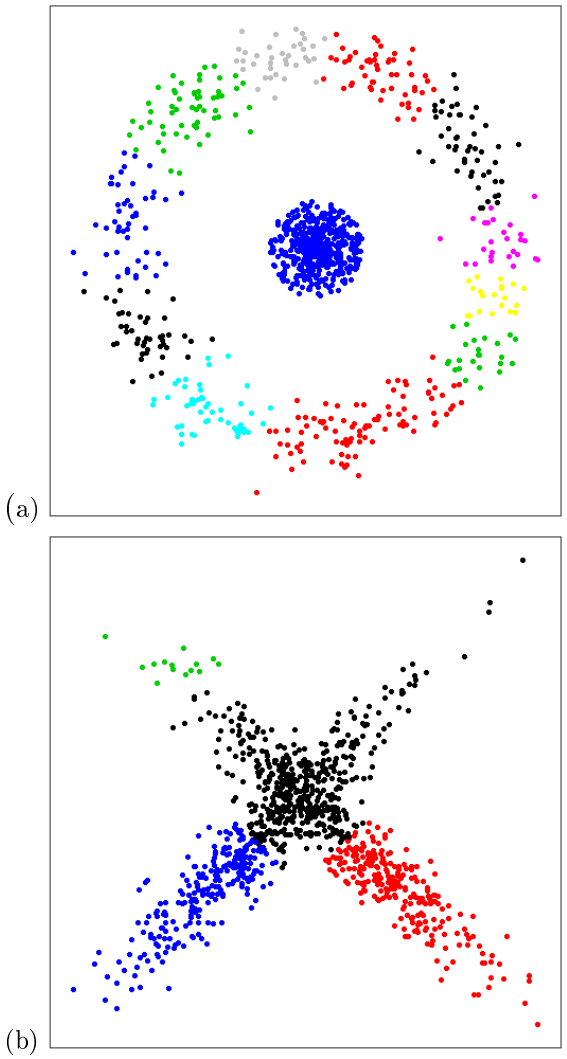


Figure B.2: Mean Shift cluster assignment for *Type C* and *Type D* datasets.

B.2 Gamma Region Affect

This section illustrates how the L parameter affects the final reassignment of observations within a region around hyperplane solutions for *Type A*, *Type C* and *Type D* datasets.

B.2.1 Mean Shift reassignment

The following figures illustrate how MS reassigned points around the hyperplane. No observations fell within the Gamma region when $L < 0.20$. This reiterates that the initial MDH solution for the *Type A* dataset was ideal. The Gamma region is highlighted in green with the final cluster label indicated as either black or red points.

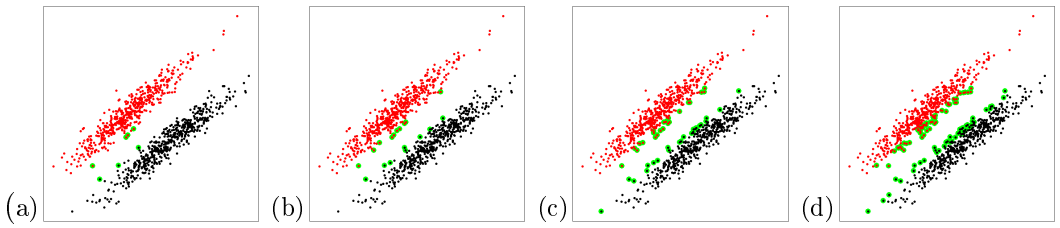


Figure B.3: Mean Shift reassignment of distinct cluster *Type A* with: $L = 0.05$ (a), $L = 0.10$ (b), $L = 0.20$ (c), $L = 0.25$ (d), $L = 0.30$ (e), $L = 0.35$ (f).

Figure B.4 illustrates how increasing L affects the final cluster solution for the *Type C* dataset. Notice that increasing Gamma resulted in MS reassigning all values to the black cluster. Figure B.5 illustrates how increasing the Gamma region affected the final cluster solution for *Type D*. Increasing the Gamma region did not change the MDH solution. This is because the MS solution for *Type D* (Figure B.2 (b)) was similar to the original MDH solution.

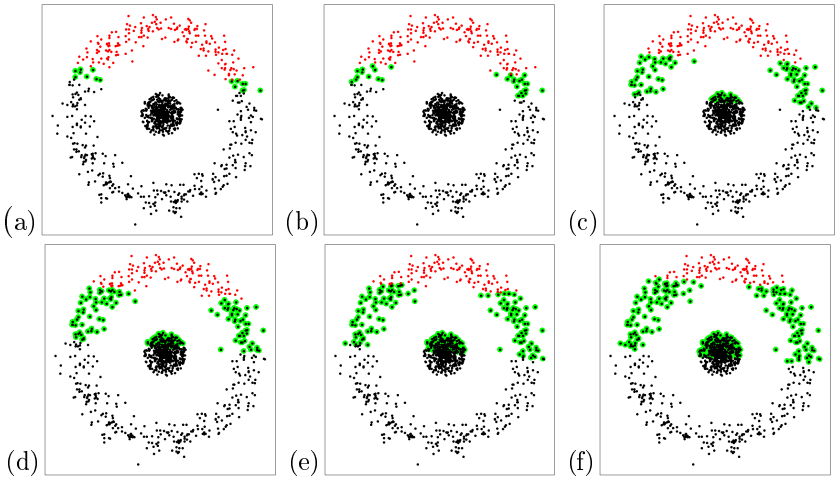


Figure B.4: Mean Shift reassignment of distinct cluster *Type C* with: $L = 0.05$ (a), $L = 0.10$ (b), $L = 0.20$ (c), $L = 0.25$ (d), $L = 0.30$ (e), $L = 0.35$ (f).

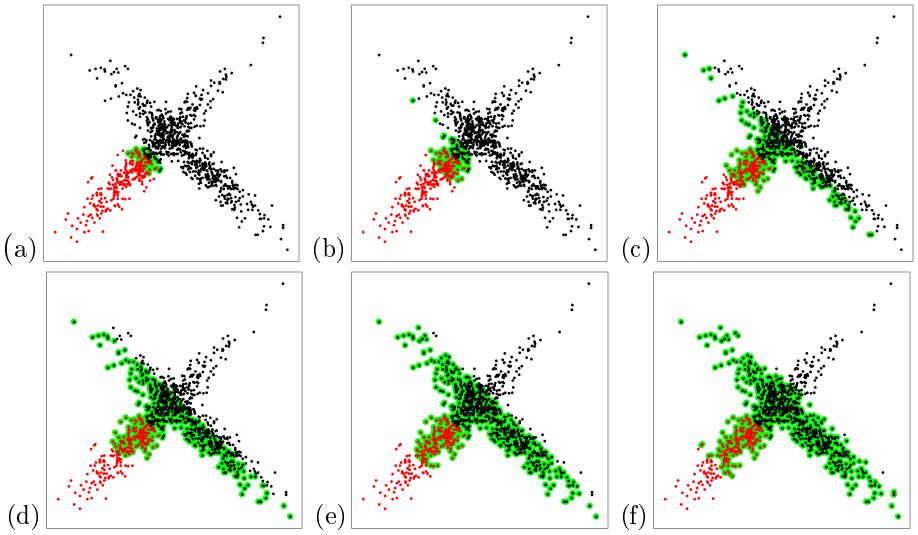


Figure B.5: Mean Shift reassignment of distinct cluster *Type D* with: $L = 0.05$ (a), $L = 0.10$ (b), $L = 0.20$ (c), $L = 0.25$ (d), $L = 0.30$ (e), $L = 0.35$ (f).

B.2.2 Single step gradient Reassignment

Applying the single step gradient approach greatly reduces the computation relative to the MS reassignment procedure. For the linearly separable data, the single step heuristic reassigned all points correctly (Figure B.6).

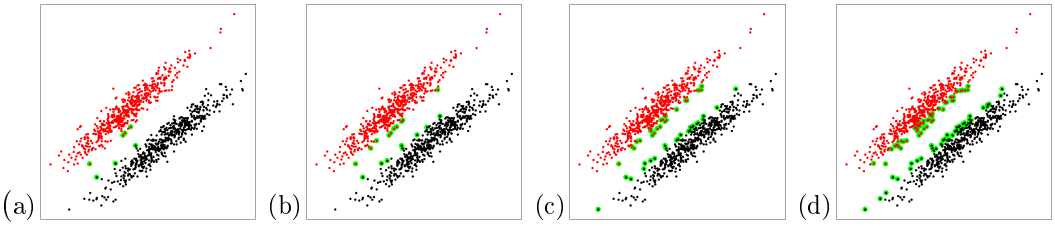


Figure B.6: Heuristic reassignment of distinct cluster *Type A* with: $L = 0.05$ (a), $L = 0.10$ (b), $L = 0.20$ (c), $L = 0.25$ (d), $L = 0.30$ (e), $L = 0.35$ (f).

Figure B.7 illustrates how the single step gradient approach reassigned the MDH *Type C* solution. At every setting of L, the heuristic approach produces errors. Increasing the Gamma region increases the number of instances that the heuristic does not conform to its formulation. We can verify this by viewing the MS *Type C* solution.

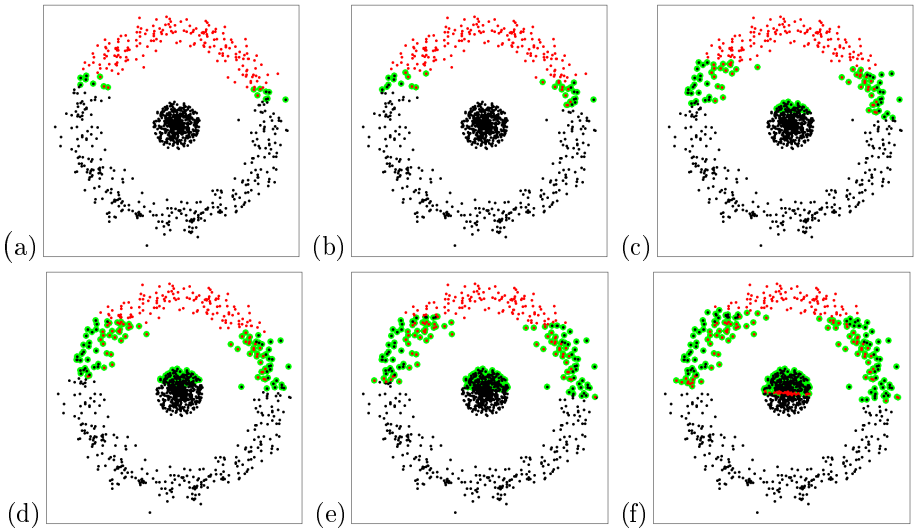


Figure B.7: Heuristic reassignment of distinct cluster *Type C* with: $L = 0.05$ (a), $L = 0.10$ (b), $L = 0.20$ (c), $L = 0.25$ (d), $L = 0.30$ (e), $L = 0.35$ (f)..

Figure B.8 illustrates how the single step gradient approach reassigned the MDH *Type D* solution. MDH nor its enhancements can effectively cluster the overlapping data *Type D*. Figure B.8 provides insight on the limitations associated with reassigning values using the heuristic approach and the impact of increasing L.

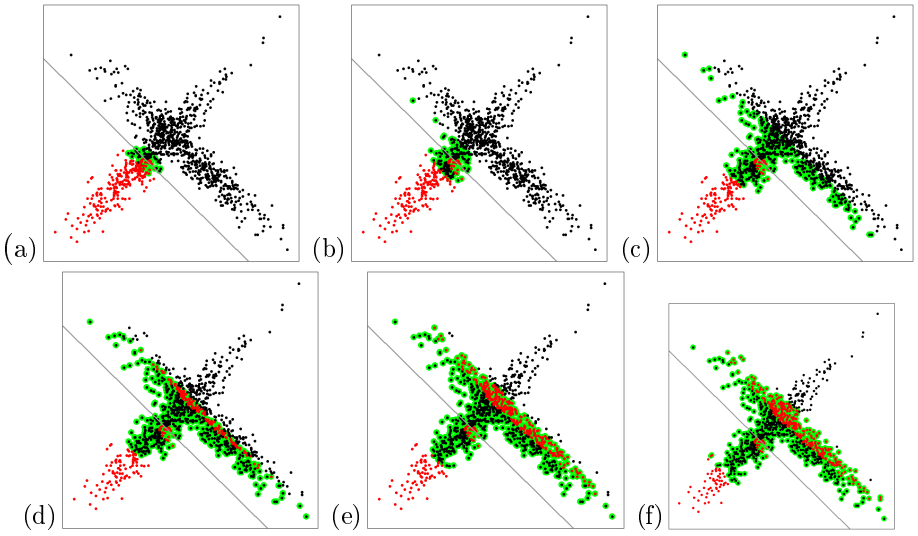


Figure B.8: Heuristic reassignment of distinct cluster *Type D* with: $L = 0.05$ (a), $L = 0.10$ (b), $L = 0.20$ (c), $L = 0.25$ (d), $L = 0.30$ (e), $L = 0.35$ (f)..

Figure B.9 illustrates the Mean Shift gradient ascent trajectories for a subset of observations from the *Type D* dataset. The original MDH solution hyperplane is indicated as a gray line. Each point is coloured to the assignment given after applying the single step gradient heuristic with $L = 0.35$. Green points represent those observations within the Gamma region while red and black indicate the final cluster assignment of each point. The two points which have gradient ascent paths indicated in yellow have initial gradients that point towards but do not converge to a mode (blue points) beyond the decision boundary. It is these instances that the heuristic does not reassign labels in accordance to its formulation (viz. the way MS would have).

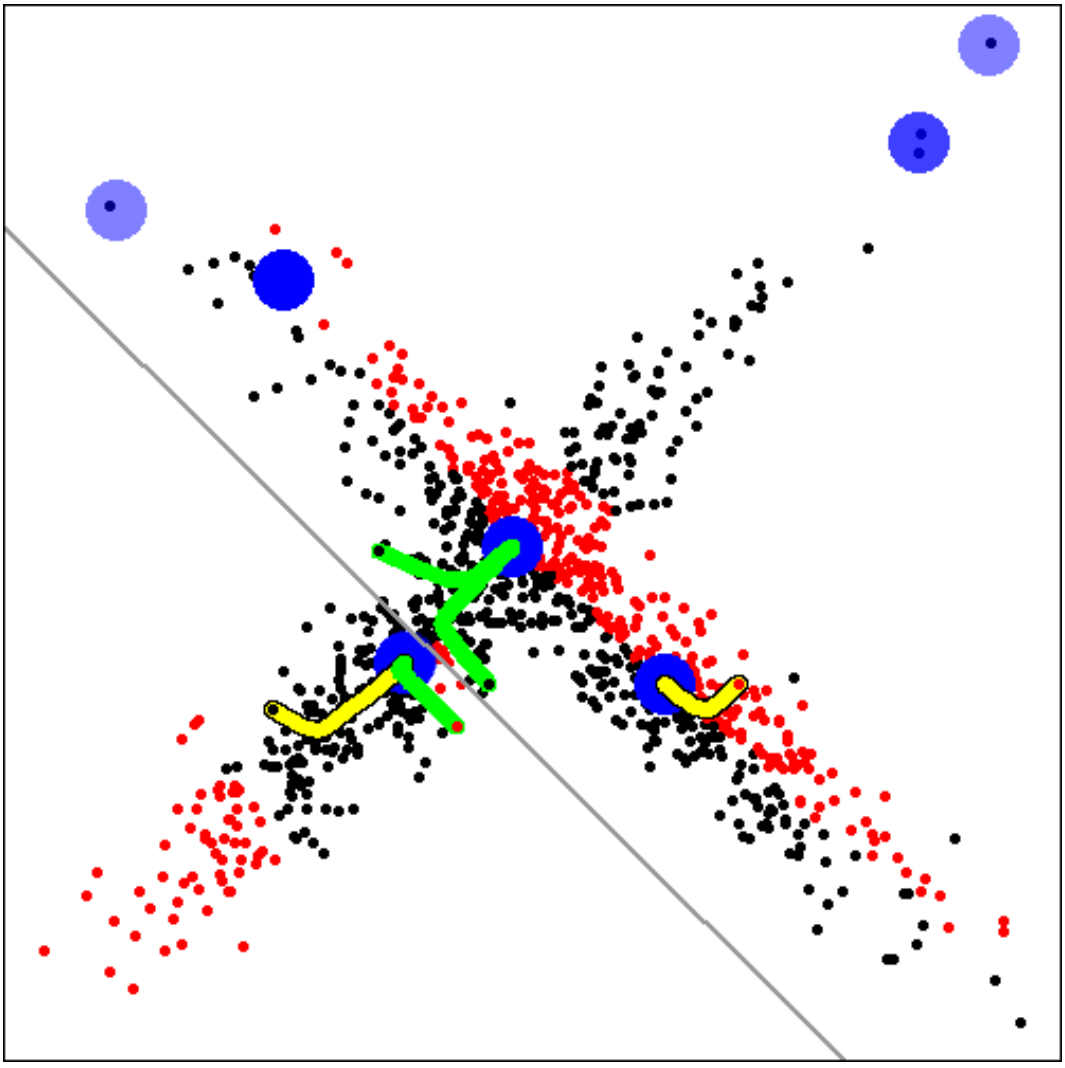


Figure B.9: Heuristic assignment of distinct cluster *Type D*. Large blue points represent the estimated modes from MS, green lines indicated gradient ascent trajectories with yellow indicating those paths which move towards but do not converge beyond the hyperplane.

B.3 Benchmark Datasets Classes

For the benchmark tests, only a subset of the data was considered such that each dataset contained only two classes. The following table indicates those classes utilised during the benchmark testing undertaken in Chapter 4.

Table B.1: Class details of benchmark datasets.

Dataset	$\mathcal{C}_1\, n$	$\mathcal{C}_2\, n$	$\mathcal{C}_1$ Name	$\mathcal{C}_2$ Name
Banknote	762	610	Class 0	Class 1
Seeds	70	70	Kama	Rosa
Wine	59	71	Class 1	Class 2
Votes	168	267	Republican	Democrat
Breast Cancer	212	357	Malignant	Benign
Synthetic Control	100	100	Normal	Cyclic

Details of benchmark datasets: size within class 1 ($\mathcal{C}_1\, n$) and class 2 ($\mathcal{C}_2\, n$) with associated labels.

B.4 Experimental Bandwidth Estimation

MDH searches for an optimal low-density separator, choosing a bandwidth that produces a density estimate with relatively few objects around the hyperplane may be ideal. The process for obtaining h_{xp} is based on the concept that an optimal bandwidth is one which yields a minimum density hyperplane with relatively few neighbouring objects. The process is as follows:

1. Consider $h = \mathrm{C} n^{-1/5} \hat{\sigma}_{pc_1}$
2. Linearly transform dataset using the first principal component vector
3. Calculate mean and standard deviation of transformed data
4. Generate a range of values for C (e.g. 0.5 to 3, where 0.9 generates heuristic)
5. Estimate the bandwidth in Step 1 according to the linear transformed data of Step 2
6. Estimate density using current bandwidth
7. Locate minimum density (within one standard deviation of the mean) and assign as decision boundary
8. Tally the number of objects in a Gamma region with a set value for L.
9. Repeat Steps 4 to 8 for all values of C
10. Choose bandwidth associated with the lowest number of objects within the Gamma region.

Appendix C

Image Segmentation

C.1 Data Structure of Images

Figure C.1 illustrates how each pixel is represented within an RGB image. Each pixel is defined by its Cartesian coordinates (x and y) and the red, green, and blue intensities which comprise its overall colour.

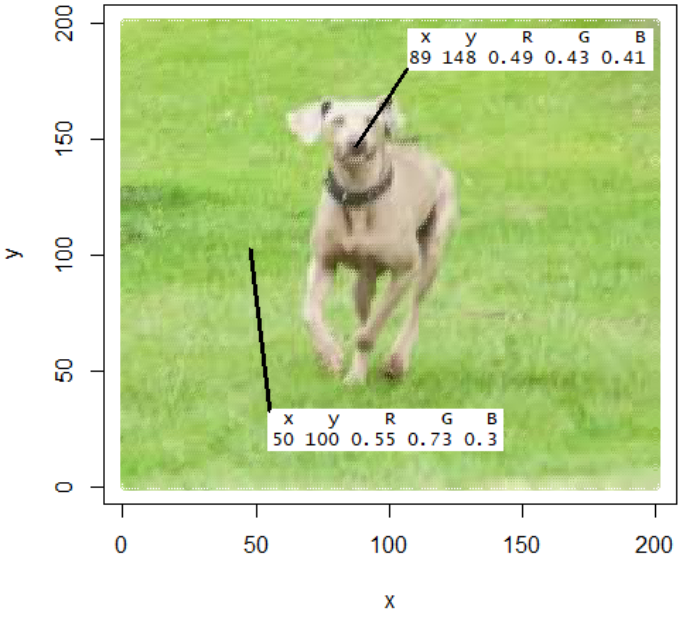
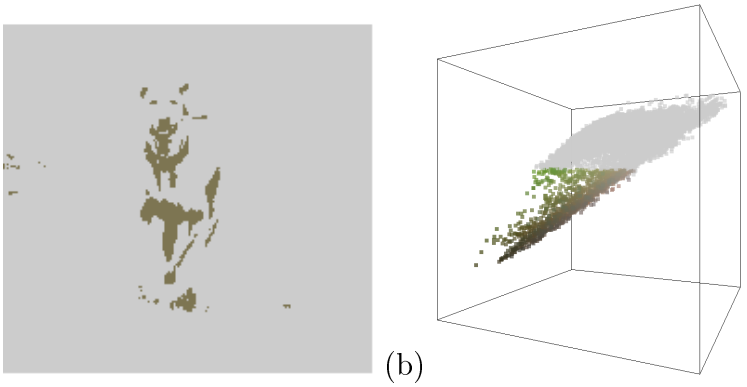


Figure C.1: Two identified pixels within the dog image with their associated x, y and RGB values.

C.2 Misclassified Portion of Dog Image

Initial MDH results clustered the image by segmenting lighter tones from darker tones. The following figure highlights the dark brown tones associated with the dog that were wrongly assigned to the background.

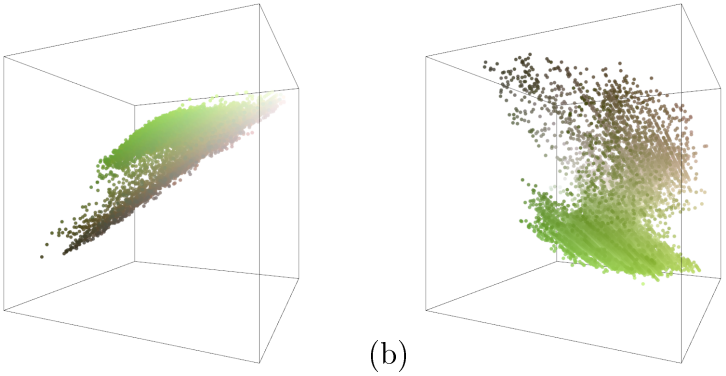


(a) (b)

Figure C.2: Subset of dog image highlighting brown tones within the image (a) and the associated scatter plot of pixels (b).

C.3 Effect of Decorrelation Stretch

Decorrelation stretch (DCS) was introduced as a tool to reduce disparity of variances along principal components and disperse pixel intensities. Figure C.3 illustrates how DCS disperses the observed pixel intensities from the dog image.



(a) (b)

Figure C.3: Pixel intensities for non-transformed (a) and decorrelated stretched (b) dog image.

C.4 Comparing Image Segmentation of Dog Image

The dog image was clustered using 2-means, Max Margin Clustering (complexity parameter set to 0.0001), MDH ($\alpha = 1.5$, $w = 5$), $\text{MDH}_{\mathbf{\Gamma}_{MS}}$ and $\text{MDH}_{\mathbf{\Gamma}_{H}}$ ($L = 0.20$). As an initial step, the image was compressed using mean prototypes from a 25×25 grid. MDH , $\text{MDH}_{\mathbf{\Gamma}_{MS}}$ and $\text{MDH}_{\mathbf{\Gamma}_{H}}$ identified the focal object within the image. The MMC and 2-means solutions contained relatively large amounts of noise.

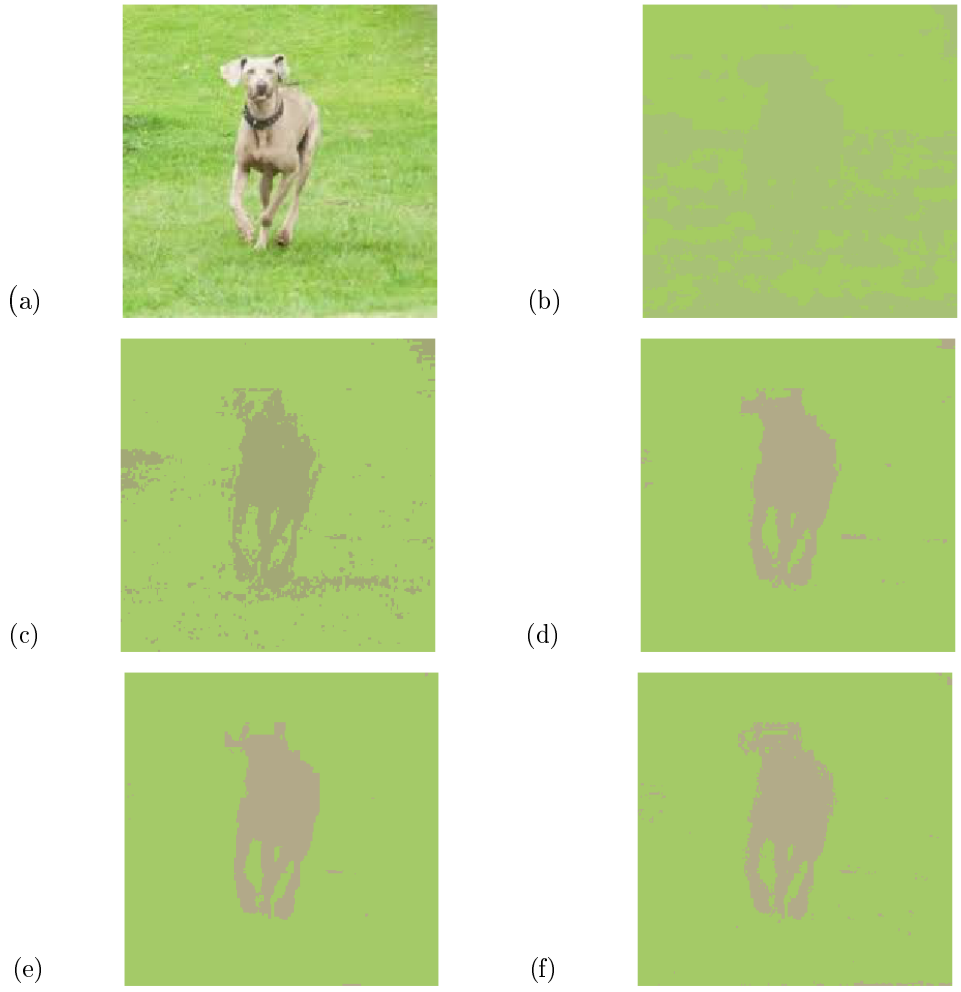


Figure C.4: Binary cluster results of Dog Image (a) using 2-means (b), Max Margin Clustering (c), Minimum Density Hyperplane Clustering (d), MDH solution reassigned by Mean Shift (e) and the single step gradient procedure (f).

List of References

Aggarwal, C.C. and Reddy, C.K. (2013). *Data clustering: algorithms and applications*. CRC press.

Agrawal, R., Gehrke, J., Gunopulos, D. and Raghavan, P. (1998). *Automatic subspace clustering of high dimensional data for data mining applications*, vol. 27. ACM.

Alley, R.E. (1999). Algorithm theoretical basis document for: decorrelation stretch.

Arthur, D. and Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In: *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035. Society for Industrial and Applied Mathematics.

Azzalini, A. and Menardi, G. (2014 April). Clustering via nonparametric density estimation: The r package pdfcluster. *Journal of Statistical Software*, vol. 57, no. 11, pp. 1–26.

Bailey, T.C. and Gatrell, A.C. (1995). *Interactive spatial data analysis*, vol. 413. Longman Scientific & Technical Essex.

Ballard, D.H. and Brown, C.M. (1982). *Computer Vision*. Prentice Hall, EngleWood Cliffs, New Jersey.

Ben-David, S., Lu, T., Pál, D. and Sotáková, M. (2009). Learning low density separators. In: *Artificial Intelligence and Statistics*, pp. 25–32.

Ben-David, S. and Von Luxburg, U. (2008). Relating clustering stability to properties of cluster boundaries. In: *COLT*, vol. 2008, pp. 379–390.

Berkhin, P. (2006). A survey of clustering data mining techniques. In: *Grouping multidimensional data*, pp. 25–71. Springer.

Carmichael, J. and Julius, R. (1968). Finding natural clusters. *Systematic Biology*, vol. 17, no. 2, pp. 144–150.

Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE transactions on pattern analysis and machine intelligence*, vol. 17, no. 8, pp. 790–799.

Comaniciu, D. and Meer, P. (1999). Mean shift analysis and applications. In: *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, pp. 1197–1203. IEEE.

Comaniciu, D., Ramesh, V. and Meer, P. (2000). Real-time tracking of non-rigid objects using mean shift. In: *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 2, pp. 142–149. IEEE.

Comaniciu, D., Ramesh, V. and Meer, P. (2001). The variable bandwidth mean shift and data-driven scale selection. In: *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 1, pp. 438–445. IEEE.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, vol. 20, no. 3, pp. 273–297.

Cox, T.F. and Cox, M.A. (2000). *Multidimensional scaling*. Chapman and hall/CRC.

Davies, E.R. (2012). *Computer and machine vision: theory, algorithms, practicalities*. Fourth edition edn. Academic Press, London.

Deselaers, T., Keysers, D. and Ney, H. (2003). Clustering visually similar images to improve image search engines. *hist*, vol. 1, no. 1, p. 1.

Dheeru, D. and Karra Taniskidou, E. (2017). UCI machine learning repository. Available at: `http://archive.ics.uci.edu/ml`

Dhillon, I.S., Guan, Y. and Kulis, B. (2004). Kernel k-means: spectral clustering and normalized cuts. In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 551–556. ACM.

Duong, T. (2018). *ks: Kernel Smoothing*. R package version 1.11.0. Available at: `https://CRAN.R-project.org/package=ks`

Duong, T. *et al.* (2007). ks: Kernel density estimation and kernel discriminant analysis for multivariate data in r. *Journal of Statistical Software*, vol. 21, no. 7, pp. 1–16.

Edwards, A.W. and Cavalli-Sforza, L.L. (1965). A method for cluster analysis. *Biometrics*, pp. 362–375.

Eisen, M.B., Spellman, P.T., Brown, P.O. and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, vol. 95, no. 25, pp. 14863–14868.

Ellegood, J., Anagnostou, E., Babineau, B., Crawley, J., Lin, L., Genestine, M., DiCicco-Bloom, E., Lai, J., Foster, J., Penagarikano, O. *et al.* (2015). Clustering autism: using neuroanatomical differences in 26 mouse models to gain insight into the heterogeneity. *Molecular psychiatry*, vol. 20, no. 1, p. 118.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X. *et al.* (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Kdd*, vol. 96, pp. 226–231.

Evers, F.T., Höppner, F., Klawonn, F., Kruse, R. and Runkler, T. (1999). *Fuzzy cluster analysis: methods for classification, data analysis and image recognition*. John Wiley & Sons.

Fisher, D. (1995). Optimization and simplification of hierarchical clusterings. In: *KDD*, pp. 118–123.

Fraley, C. and Raftery, A.E. (1998). How many clusters? which clustering method? answers via model-based cluster analysis. *The computer journal*, vol. 41, no. 8, pp. 578–588.

Friedman, J., Hastie, T. and Tibshirani, R. (2001). *The elements of statistical learning*. 10. Springer series in statistics New York, NY, USA:.

Fukunaga, K. and Hostetler, L. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory*, vol. 21, no. 1, pp. 32–40.

Gelder, M., Gath, D. and Mayou, R. (1989). *Oxford textbook of psychiatry*. Oxford university press.

Gersho, A. and Gray, R.M. (1991). *Vector quantization and signal compression*, vol. 159. Springer Science & Business Media.

Gould, M.S., Petrie, K., Kleinman, M.H. and Wallenstein, S. (1994). Clustering of attempted suicide: New zealand national data. *International Journal of Epidemiology*, vol. 23, no. 6, pp. 1185–1189.

Guha, S., Rastogi, R. and Shim, K. (1998). Cure: an efficient clustering algorithm for large databases. In: *ACM Sigmod Record*, vol. 27, pp. 73–84. ACM.

Halkidi, M., Batistakis, Y. and Vazirgiannis, M. (2001*a*). On clustering validation techniques. *Journal of intelligent information systems*, vol. 17, no. 2-3, pp. 107–145.

Halkidi, M., Batistakis, Y. and Vazirgiannis, M. (2001*b*). On clustering validation techniques. *Journal of intelligent information systems*, vol. 17, no. 2-3, pp. 107–145.

Hartigan, J.A. (1975). *Clustering Algorithms*. Wiley, New York.

Hartigan, J.A. and Wong, M.A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108.

Hinneburg, A., Keim, D.A. *et al.* (1998). An efficient approach to clustering in large multimedia databases with noise. In: *KDD*, vol. 98, pp. 58–65.

Hirschberg, J.G., Maasoumi, E. and Slottje, D.J. (1991). Cluster analysis for measuring welfare and quality of life across countries. *Journal of econometrics*, vol. 50, no. 1-2, pp. 131–150.

Hofmeyr, D. (2018). *PPCI: Projection Pursuit for Cluster Identification*. R package version 0.1.1.
Available at: `https://CRAN.R-project.org/package=PPCI`

Hofmeyr, D.P. and Pavlidis, N.G. (2018). *PPCI: an R Package for Cluster Identification using Projection Pursuit*. Unpublished.

Huang, Z. (1997). Clustering large data sets with mixed numeric and categorical values. In: *Proceedings of the 1st pacific-asia conference on knowledge discovery and data mining,(PAKDD)*, pp. 21–34. Singapore.

Huber, P.J. (1985). Projection pursuit. *The annals of Statistics*, pp. 435–475.

James, G., Witten, D., Hastie, T. and Tibshirani, R. (2013). *An introduction to statistical learning*, vol. 112. Springer.

Jolliffe, I. (2011). Principal component analysis. In: *International encyclopedia of statistical science*, pp. 1094–1096. Springer.

Karypis, G., Han, E.-H. and Kumar, V. (1999). Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, vol. 32, pp. 68–75.

Kassambara, A. (2017). Practical guide to cluster analysis in r. *CreateSpace: North Charleston, SC, USA*.

Kumar, M.S. (2017). A survey on clustering algorithms used to perform image segmentation.

Linden, G., Smith, B. and York, J. (2003). Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, , no. 1, pp. 76–80.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137.

MacQueen, J. *et al.* (1967). Some methods for classification and analysis of multivariate observations. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, pp. 281–297. Oakland, CA, USA.

Martin, D., Fowlkes, C., Tal, D. and Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2, pp. 416–423. IEEE.

Moore, W.C., Meyers, D.A., Wenzel, S.E., Teague, W.G., Li, H., Li, X., D'Agostino Jr, R., Castro, M., Curran-Everett, D., Fitzpatrick, A.M. *et al.* (2010). Identification of asthma phenotypes using cluster analysis in the severe asthma research program. *American journal of respiratory and critical care medicine*, vol. 181, no. 4, pp. 315–323.

Parker, J.R. (2010). *Algorithms for Image Processing and Computer Vision*. Second edition edn. John Wiley & Sons, Indianapolis.

Pavlidis, N.G., Hofmeyr, D.P. and Tasoulis, S.K. (2015). Minimum density hyperplane: An unsupervised and semi-supervised classifier. *stat*, vol. 1050, p. 15.

Pavlidis, N.G., Hofmeyr, D.P. and Tasoulis, S.K. (2016). Minimum density hyperplanes. *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 5414–5446.

Prados, J. (2018). *bmrm: Bundle Methods for Regularized Risk Minimization Package*. R package version 3.7.
Available at: `https://CRAN.R-project.org/package=bmrm`

Punj, G. and Stewart, D.W. (1983). Cluster analysis in marketing research: Review and suggestions for application. *Journal of marketing research*, pp. 134–148.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
Available at: `https://www.R-project.org/`

Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C.-H., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J.P. *et al.* (2001). Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences*, vol. 98, no. 26, pp. 15149–15154.

Rinaldo, A. and Wasserman, L. (2010 Novemeber). Generalized density clustering. *The Annals of Statistics*, vol. 38, no. 5, pp. 2678–2722.

Rousseeuw, P.J. and Kaufman, L. (1990). Finding groups in data. *Series in Probability & Mathematical Statistics*, pp. 111–112.

Schölkopf, B., Smola, A. and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, vol. 10, no. 5, pp. 1299–1319.

Silverman, B.W. (1986). *Density estimation for statistics and data analysis*, vol. 26. CRC press.

Sokal, R.R. (1963). The principles and practice of numerical taxonomy. *Taxon*, pp. 190–199.

Struyf, A., Hubert, M., Rousseeuw, P. *et al.* (1997). Clustering in an object-oriented environment. *Journal of Statistical Software*, vol. 1, no. 4, pp. 1–30.

Tan, P.-N., Steinbach, M. and Kumar, V. (2013). Data mining cluster analysis: basic concepts and algorithms. *Introduction to data mining*.

Tasoulis, S.K., Tasoulis, D.K. and Plagianakos, V.P. (2010). Enhancing principal direction divisive clustering. *Pattern Recognition*, vol. 43, no. 10, pp. 3391–3411.

Wang, W., Yang, J., Muntz, R. *et al.* (1997). Sting: A statistical information grid approach to spatial data mining. In: *VLDB*, vol. 97, pp. 186–195.

Ward Jr, J.H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, vol. 58, no. 301, pp. 236–244.

Wilks, D.S. (2011). Cluster analysis. In: *International geophysics*, vol. 100, pp. 603–616. Elsevier.

Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Philip, S.Y. *et al.* (2008). Top 10 algorithms in data mining. *Knowledge and information systems*, vol. 14, no. 1, pp. 1–37.

Xu, L., Neufeld, J., Larson, B. and Schuurmans, D. (2005). Maximum margin clustering. In: *Advances in neural information processing systems*, pp. 1537–1544.

Xu, L. and Schuurmans, D. (2005). Unsupervised and semi-supervised multi-class support vector machines. In: *AAAI*, vol. 5, p. 13.

Xu, R. and Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on neural networks*, vol. 16, no. 3, pp. 645–678.

Yates, K.R. and Pavlidis, N.G. (2016). Minimum density hyperplanes in the feature space. In: *Big Data (Big Data), 2016 IEEE International Conference on*, pp. 3613–3618. IEEE.

Zaki, M.J., Meira Jr, W. and Meira, W. (2014). *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press.

Zhang, T., Ramakrishnan, R. and Livny, M. (1996). Birch: an efficient data clustering method for very large databases. In: *ACM Sigmod Record*, vol. 25, pp. 103–114. ACM.